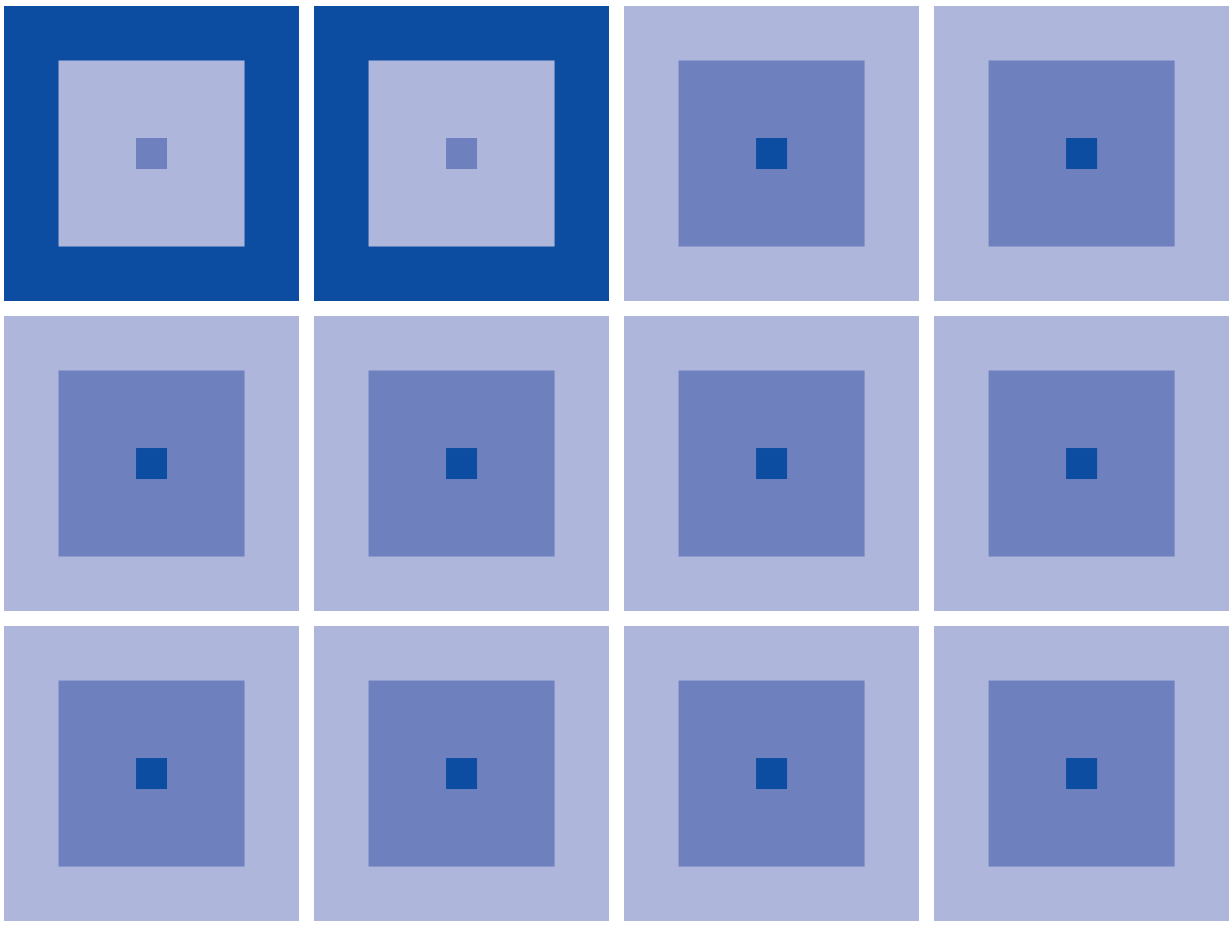
MF598-04

EPSON

CMOS 4-BIT SINGLE CHIP MICROCOMPUTER

S1C62 Family

Development Tool Reference Manual



セイコーエプソン株式会社

本資料のご使用につきましては、次の点にご留意願います。

1. 本資料の内容については、予告なく変更することがあります。
2. 本資料の一部、または全部を弊社に無断で転載、または、複製など他の目的に使用することは堅くお断りします。
3. 本資料に掲載される応用回路、プログラム、使用方法等はあくまでも参考情報であり、これらに起因する第三者の権利(工業所有権を含む)侵害あるいは損害の発生に対し、弊社は如何なる保証を行うものではありません。また、本資料によって第三者または弊社の工業所有権の実施権の許諾を行うものではありません。
4. 特性表の数値の大小は、数直線上の大小関係で表しています。
5. 本資料に掲載されている製品のうち、「外国為替および外国貿易法」に定める戦略物資に該当するものについては、輸出する場合、同法に基づく輸出許可が必要です。
6. 本資料に掲載されている製品は、一般民生用です。生命維持装置その他、きわめて高い信頼性が要求される用途を前提としていません。よって、弊社は本(当該)製品をこれらの用途に用いた場合の如何なる責任についても負いかねます。

MS-DOS、Windows、Windows95、Windows98およびWindows NTは米国マイクロソフト社の登録商標です。
PC/AT、PS/2、PC-DOS、VGA、EGAおよびIBMは、米国International Business Machines社の登録商標です。
NEC PC-9800シリーズおよびNECは日本電気株式会社の登録商標です。
その他のブランド名または製品名は、それらの所有者の商標もしくは登録商標です。

© **SEIKO EPSON CORPORATION** 2001

S1C62 Family Development Tool Reference Manual

はじめに

本書はCMOS 4-bit Single Chip Microcomputer S1C62 Familyの開発支援ツールの概要と操作を以下のパートに分けて説明します。

I. イントロダクション
II. Development Tool Management System DMS6200
III. Cross Assembler ASM62XX
IV. Melody Assembler MLA628X
V. Function Option Generator FOG62XX
VI. Segment Option Generator SOG62XX
VII. Evaluation Board S5U1C62XXXE
VIII. ICE Control Software ICS62XX
IX. Mask Data Checker MDC62XX

お読みになる前に

本書では S1C62 Family の各機種に共通な内容を説明しているため、機種名を"S1C62XXX"、ソースファイルや出力ファイルの名称を"C2XXYYY"と記述しています。"XXX"の部分を各機種名に置き換えてお読みください。また、"YYY"の部分はお客さまごとにセイコーエプソンが指定させていただきます。

例: 開発機種がS1C6S460で"YYY"の部分を"0A0"と指定させていただいた場合

S1C6XXXX → S1C6S460
CXXXYYY → CS460A0

参照マニュアル

各機種固有の内容、デバイスの詳細などは以下のマニュアルで解説されています。必要に応じて参照してください。

開発ツールについて ☞ S5U1C62xxxD Manual (Development Software Tool for S1C62xxx)
S5U1C62xxxE Manual (Evaluation Board for S1C62xxx)
S5U1C62000H Manual (S1C60/62 Family In-Circuit Emulator)

デバイス(S1C62xxx)について ☞ S1C62xxxテクニカルマニュアル

インストラクションについて ☞ S1C6200/6200AコアCPUマニュアル

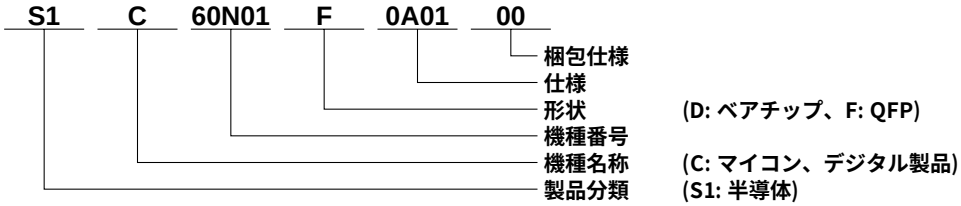
※ 本書の"ICE"はS5U1C62000Hを、"Evaluation Board"はS5U1C62xxxEを表します。

製品型番変更のご案内

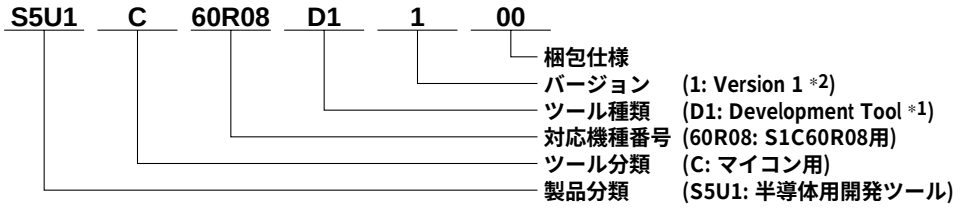
2001年4月1日より、弊社半導体製品の製品型番が以下のとおり変更となりますので、4月1日以降のご発注につきましては変更後の製品型番にてお願い申し上げます。
なお、製品型番の詳細仕様につきましては、弊社営業担当にお問い合わせください。

製品型番体系

●デバイス



●開発ツール



*1: ツールの種類は、新旧型番対応表を参照してください。(マニュアル類には一桁で記載されているものもあります。)
*2: マニュアル類には、実際のバージョンは記載されておりません。

新旧型番対応表

●S1C60ファミリ

旧型番	新型番
E0C6001	S1C60N01
E0C6002	S1C60N02
E0C6003	S1C60N03
E0C6004	S1C60N04
E0C6005	S1C60N05
E0C6006	S1C60N06
E0C6007	S1C60N07
E0C6008	S1C60N08
E0C6009	S1C60N09
E0C6011	S1C60N11
E0C6013	S1C60N13
E0C6014	S1C60140
E0C60R08	S1C60R08

●S1C62ファミリ

旧型番	新型番
E0C621A	S1C621A0
E0C6215	S1C62150
E0C621C	S1C621C0
E0C6S27	S1C6S2N7
E0C6S37	S1C6S3N7
E0C623A	S1C6N3A0
E0C623E	S1C6N3E0
E0C6S32	S1C6S3N2
E0C6233	S1C62N33
E0C6235	S1C62N35
E0C623B	S1C6N3B0
E0C6244	S1C62440
E0C624A	S1C624A0
E0C6S46	S1C6S460

旧型番	新型番
E0C6247	S1C62470
E0C6248	S1C62480
E0C6S48	S1C6S480
E0C624C	S1C624C0
E0C6251	S1C62N51
E0C6256	S1C62560
E0C6292	S1C62920
E0C6262	S1C62N62
E0C6266	S1C62660
E0C6274	S1C62740
E0C6281	S1C62N81
E0C6282	S1C62N82
E0C62M2	S1C62M20
E0C62T3	S1C62T30

開発ツール新旧型番対応表

●S1C60/62ファミリ関係の開発ツール

旧型番	新型番
ASM62	S5U1C62000A
DEV6001	S5U1C60N01D
DEV6002	S5U1C60N02D
DEV6003	S5U1C60N03D
DEV6004	S5U1C60N04D
DEV6005	S5U1C60N05D
DEV6006	S5U1C60N06D
DEV6007	S5U1C60N07D
DEV6008	S5U1C60N08D
DEV6009	S5U1C60N09D
DEV6011	S5U1C60N11D
DEV60R08	S5U1C60R08D
DEV621A	S5U1C621A0D
DEV621C	S5U1C621C0D
DEV623B	S5U1C623B0D
DEV6244	S5U1C62440D
DEV624A	S5U1C624A0D
DEV624C	S5U1C624C0D
DEV6248	S5U1C62480D
DEV6247	S5U1C62470D

旧型番	新型番
DEV6262	S5U1C62620D
DEV6266	S5U1C62660D
DEV6274	S5U1C62740D
DEV6292	S5U1C62920D
DEV62M2	S5U1C62M20D
DEV6233	S5U1C62N33D
DEV6235	S5U1C62N35D
DEV6251	S5U1C62N51D
DEV6256	S5U1C62560D
DEV6281	S5U1C62N81D
DEV6282	S5U1C62N82D
DEV6S27	S5U1C6S2N7D
DEV6S32	S5U1C6S3N2D
DEV6S37	S5U1C6S3N7D
EVA6008	S5U1C60N08E
EVA6011	S5U1C60N11E
EVA621AR	S5U1C621A0E2
EVA621C	S5U1C621C0E
EVA6237	S5U1C62N37E
EVA623A	S5U1C623A0E

旧型番	新型番
EVA623B	S5U1C623B0E
EVA623E	S5U1C623E0E
EVA6247	S5U1C62470E
EVA6248	S5U1C62480E
EVA6251R	S5U1C62N51E1
EVA6256	S5U1C62N56E
EVA6262	S5U1C62620E
EVA6266	S5U1C62660E
EVA6274	S5U1C62740E
EVA6281	S5U1C62N81E
EVA6282	S5U1C62N82E
EVA62M1	S5U1C62M10E
EVA62T3	S5U1C62T30E
EVA6S27	S5U1C6S2N7E
EVA6S32R	S5U1C6S3N2E2
ICE62R	S5U1C62000H
KIT6003	S5U1C60N03K
KIT6004	S5U1C60N04K
KIT6007	S5U1C60N07K

I

S1C62 Family Development Tool

イントロダクション

ここでは4-bit Single Chip Microcomputer S1C62XXX用開発支援ツールの構成と開発環境について説明します。

イントロダクション

- 目　次 -

1 開発支援ツールの種類 I-1
1.1 ソフトウェアツールS5U1C62xxxDの構成 I-1
1.2 ハードウェアツールの構成 I-1
2 開発環境 I-1
3 開発フロー I-2
4 インストール I-3
5 機種による相違点と注意事項 I-4
6 "故障かな?"と思う前に (ソフトウェア開発ツール) I-4

1 開発支援ツールの種類

ここでは、開発支援ツールのソフトウェアおよびハードウェアの構成について説明します。

1.1 ソフトウェアツールS5U1C62xxxDの構成

S5U1C62xxxDは各機種ごとに用意されているソフトウェア開発支援ツールで、以下のソフトウェアが含まれています。

1. Development Tool Management System DMS6200 .... 各ソフトウェアのメニュー選択/起動ソフトウェア
2. Cross Assembler ASM62XX ..... プログラム作成クロスアセンブラ
3. Melody Assembler MLA628X(注) ..... メロディデータ作成プログラム
4. Function Option Generator FOG62XX ..... ファンクションオプションデータ作成プログラム
5. Segment Option Generator SOG62XX(注) ..... セグメントオプションデータ作成プログラム
6. ICE Control Software ICS62XX ..... ICE制御プログラム
7. Mask Data Checker MDC62XX ..... マスクデータ作成プログラム

注: 3のMelody Assembler MLA628Xはメロディ機能を持つ機種(S1C62N8X)にのみ設定されています。
5のSegment Option Generator SOG62XXはLCDドライバとセグメントオプションを持つ機種にのみ設定されています。

1.2 ハードウェアツールの構成

ハードウェア開発支援ツールとしては以下の2種類が全機種に用意されています。

1. In-Circuit Emulator S5U1C62000H .... 高度なデバッグが可能なインサーキットエミュレータ(各機種共通)
2. Evaluation Board S5U1C62xxxE ..... 実ICと同等の機能を持つエバリュエーションボード(各機種別)

2 開発環境

S1C62XXXのアプリケーション開発は、基本的に以下のホストシステムにより行います。

- IBM社製 PC/AT (PC-DOS Ver. 2.0以上)

お客さまがS1C62XXXシリーズを開発する際は、セイコーエプソンより標準でサポートされる開発ツール以外に、上記のホストコンピュータおよびエディタ、必要に応じてP-ROMライタ、プリンタなどを準備する必要があります。

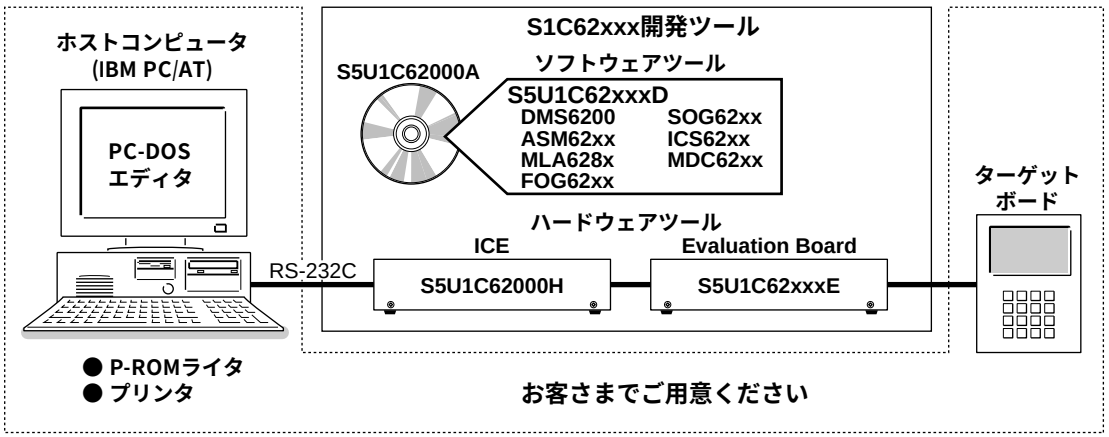
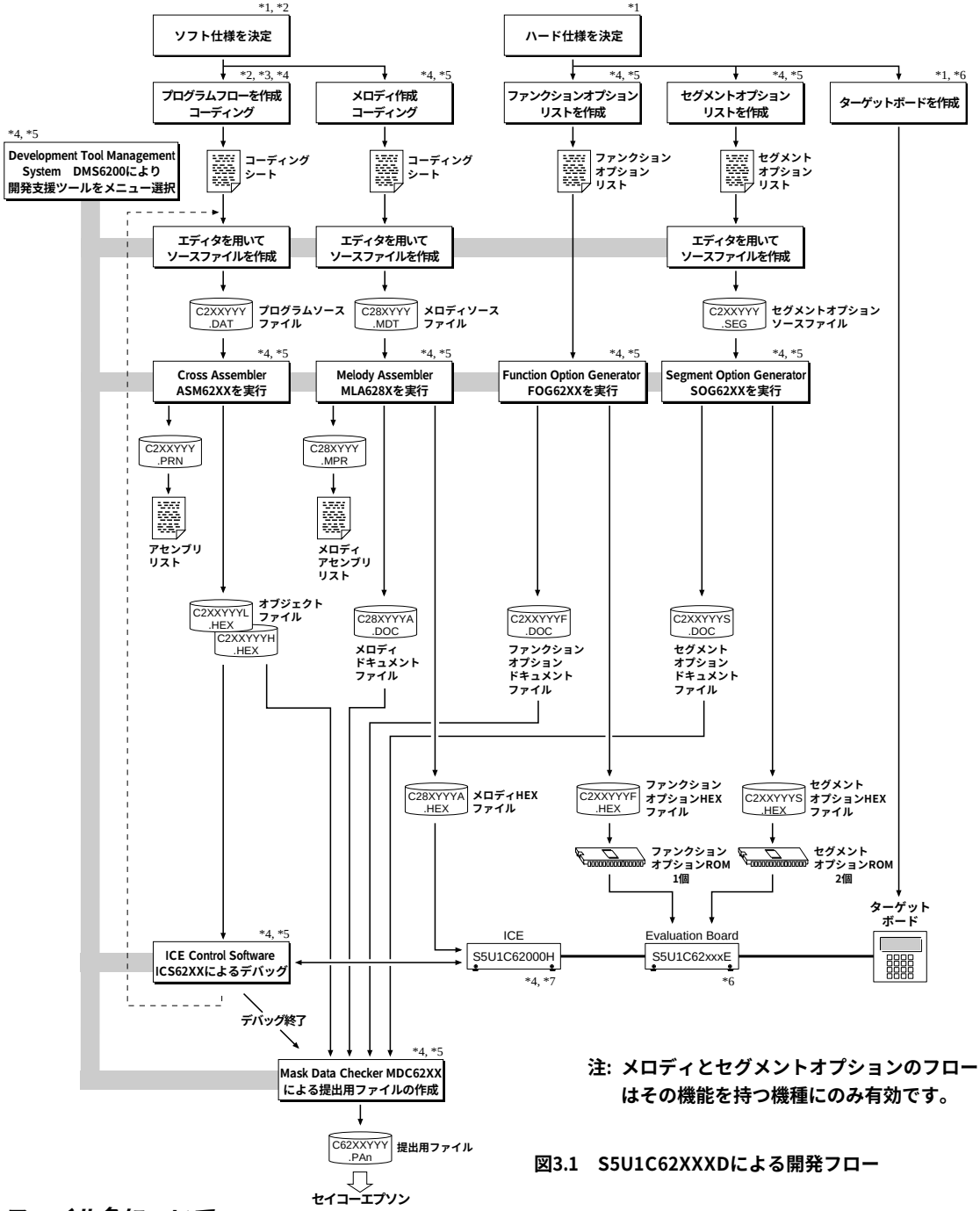


図2.1 システム構成

注: ホストコンピュータのメモリ容量として最低140Kバイト以上のユーザーRAMが必要です。
ICE(S5U1C62000H)はRS-232Cシリアルインタフェースでホストコンピュータと接続しますので、ホストコンピュータによっては非同期通信用アダプタボードが必要となります。

3 開発フロー

S5U1C62xxxDによる開発フローを以下に示します。



注: メロディとセグメントオプションのフローはその機能を持つ機種にのみ有効です。

図3.1 S5U1C62XXXDによる開発フロー

ファイル名について

各開発支援ツールの入出力ファイルの名称は、すべて"C2XXYYY"を共通に使用します。基本的にこの形式で各ファイルを作成してください。"YYY"はお客さまごとにセイコーエプソンが指定させていただきます。

☞ 参照マニュアル

*1 S1C62xxxテクニカルマニュアル(Hardware)
*2 S1C62xxxテクニカルマニュアル(Software)
*3 S1C6200/6200AコアCPUマニュアル
*4 S1C62 Family Development Tool Reference Manual (本マニュアル)
*5 S5U1C62xxxD Manual
*6 S5U1C62xxxE Manual
*7 S5U1C62000H Manual

4 インストール

S5U1C62xxxDの各ツールはS5U1C62000A(S1C60/62 Familyアセンブラパッケージ)のCD-ROMに収められており、インストールはCD-ROM内のインストーラ(Setup.exe)により行います。
インストール方法については、"S5U1C62000A Manual"を参照してください。

注: DMS6200はカレントディレクトリにあるファイルからメニューを作成します。したがって、各ツールをDMS6200があるディレクトリから移動しないでください。また、エディタ(DOS版)等をDMS6200から起動させるには、その実行ファイルをDMS6200のディレクトリにコピーしてください。

5 機種による相違点と注意事項

S5U1C62xxxDに含まれるソフトウェアツールのなかで、以下の2種類については機種により存在しない場合があります。

(1) Segment Option Generator SOG62XX
セグメントオプションが設定されていない機種のソフトウェアツールには含まれません。

(2) Melody Assembler MLA628X
メロディ機能を持たない機種(S1C62N8X以外)のソフトウェアツールには含まれません。

ホストシステムの設定においては、以下の点に注意してください。

(1) ホストコンピュータのメモリ容量として最低140Kバイト以上のユーザーRAMが必要です。

(2) ICE(S5U1C62000H)はRS-232Cシリアルインタフェースでホストコンピュータと接続しますので、ホストコンピュータによっては非同期通信用アダプタボードが必要となります。

(3) Mask Data Checker MDC62XXは多くのファイルを扱うため、CONFIG.SYSに記述するファイル数の設定を10以上としてください。

6 "故障かな?"と思う前に (ソフトウェア開発ツール)

ツール	症 状	対 策
ICE S5U1C62000H	起動画面直後、表示されないまたは操作ができない。	以下の点を確認し、正しい設定を行ってください。 ・ RS-232Cケーブルは正しく接続されていますか? ・ RSDRV.SYSが設定されていますか? ・ MODE.COMはありますか? ・ 実行ファイルは正しいものですか? PC-DOS ICS62XXW.EXE ・ DOSのバージョンはあっていますか? PC-DOS Ver. 2.1以上 ・ ICE本体のボーレート設定用のDIP SWは正しく設定されていますか? ・ ICEのヒューズは切れていませんか?
	起動直後にICEのヒューズが切れた。	以下の点を確認し、正しい設定を行ってください。 ・ Evaluation Boardと接続しているF1、F5コネクタが逆になっていませんか? ・ 接続しているターゲットボード上で電源がショートしていませんか?
	起動直後 <ILLEGAL VERSION ICE6200> と表示される。	ICEのバージョンがあっていません。最新のものと取り替えてください。
	起動直後 <ILLEGAL VERSION PARAMETER FILE> と表示される。	ICS62XXP.PARのバージョンがあっていません。最新のものと取り替えてください。

ツール	症　状	対　策
ICE **S5U1C62000H**	"A"コマンドによりイミディエート値A(10)、B(11)が正常に入力できない。	A、B入力はAレジスタ、Bレジスタとして予約されています。A(10)、B(11)を入力する場合は0A、0Bと記述してください。 **例:** LD A,B　Bレジスタ内容がAレジスタにロードされます。 LD B,0A　イミディエート値AがBレジスタにロードされます。
	"SD"コマンドにて<UNUSED AREA>と表示される。	データを書き込んだ次のアドレスが未使用領域の場合出力されますが、問題ありません。 リードオンリーエリア以外はデータは正しくセットされます。
	ブレークトレースモードを指定するとリアルタイムに実行しない。	ブレーク条件に一致した場合、CPUは一時停止しますのでリアルタイムに実行しません。
	ICEコマンドにてBuzzerおよびFout出力のI/Oメモリにデータを書き込んだ場合、Evaluation Boardより出力されない。	リアルタイムランのみ出力が可能となります。
SOG62XX	セグメントソースファイル上でアドレスが正しく設定されているのに"Rエラー"となる。	以下の点を確認し、正しい設定を行ってください。 ・ アドレスは大文字で表記されていますか? ・ 出力ポートは2端子ごとの設定になっていますか?
ASM62XX	最終ページを越えているにもかかわらず"Rエラー"となる。	ページを越えるごとに"Rエラー"が出力される仕様となっています。ページが変わる場合にはORG、PAGE等のメモリ設定疑似命令が必要です。 クロスアセンブラマニュアルの"メモリ設定疑似命令"の項を参照してください。
MDC62XX	起動しない。	以下の点を確認し、正しい設定を行ってください。 ・ OSの環境設定ファイル"CONFIG.SYS"上でファイル数の設定は10以上になっていますか?
MLA628X	Melodyが出力されない。	以下の点を確認し、正しい設定を行ってください。 ・ ICEとEvaluation Boardを接続している場合、ICEの"OPTLDコマンド"は実行していますか? ・ Evaluation Board単独で使用している場合、MELODY ROMは装着されていますか? ・ メロディデータのアタックビットは"1"になっていますか?
Evaluation Board **S5U1C62xxxE**	Evaluation Board単体使用時に動作しない。	以下の点を確認し、正しい設定を行ってください。 ・ F.HEXおよびS.HEXのEP-ROMをターゲット用のEP-ROMに交換しましたか? ・ F.HEXおよびS.HEXのEP-ROMは正しく装着されていますか? ・ 電源電圧は正しく供給されていますか? (DC5V 3A以上) ・ プログラムROMのHとLが逆になっていませんか? ・ プログラムROMに27C256を使用した場合、データは4000番地から書き込まれていますか? ・ Evaluation Board上面のEN/DISスイッチがEN側に倒れていませんか?
	ターゲットのセグメントが点灯しない。	以下の点を確認し、正しい設定を行ってください。 ・ S.HEXのEP-ROMはアクセスタイム170ns以下のものを使用していますか? ・ Evaluation Board上面カバー内部のVADJボリュームが絞られていませんか?

II

Development Tool Management System

DMS6200

ここではソフトウェアツールのメニュー選択による
起動を行うDevelopment Tool Management System
DMS6200の操作方法について説明します。

Development Tool Management System DMS6200

- 目 次 -

1 機種による相違点 *II-1*

2 DMS6200の概要 *II-1*

3 DMS6200の操作方法 *II-1*

1 機種による相違点

DMS6200はS1C62 Familyの全機種に共通のソフトウェアツールで、操作方法に違いはありません。
ただし、各機種のソフトウェアツールの構成の違いやDMS6200が存在するディレクトリの内容の違いによりメニュー画面等の内容が異なります。

説明および表示画面例に記載された以下の2種類については、機種により存在しない場合があります。

(1) SOG62XX、C2XXYYYS.*はセグメントオプションを持つ機種にのみ存在します。

(2) MLA628X、C28XYYY.M*およびC28XYYYA.*はメロディ機能を持つ機種にのみ存在します。

上記の機能を持たない機種を使用される場合は、説明中に記載のそれぞれのプログラム名およびファイル名を無視してください。

S5U1C62xxxDに含まれるソフトウェアツールについては"S5U1C62xxxD Manual"を参照してください。

2 DMS6200の概要

DMS6200(Development Tool Management System)は、S5U1C62xxxDの開発支援ツールおよびエディタなどをメニュー形式で起動することができます。これにより、デバッグ時に頻繁に行われる各ソフトウェアの起動が効率よく行えます。
図2.1にDMS6200による各ソフトウェアの実行フローを示します。

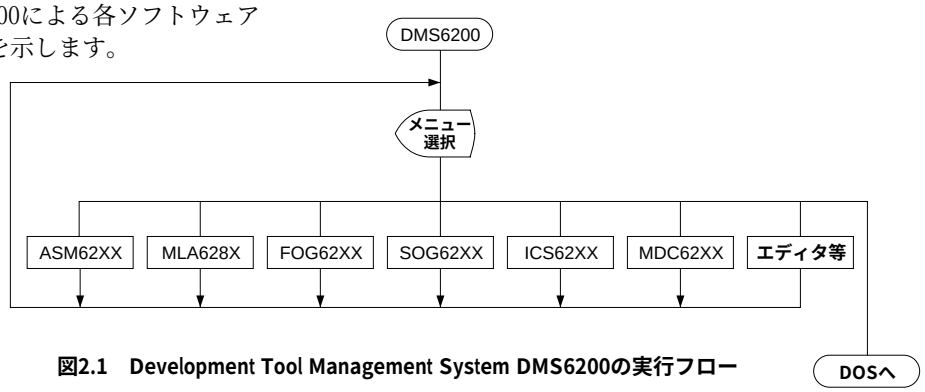


図2.1 Development Tool Management System DMS6200の実行フロー

3 DMS6200の操作方法

DMS6200を起動させる前に各ソフトウェア開発支援ツールが入っているディレクトリをカレントディレクトリに設定してください。
開発支援ツールには、それぞれ入力ファイル(ソースファイルなど)が必要ですので、各支援ツールの説明を参照し入力ファイルを作成後、下記の操作を行ってください。

(1) 実行ディスク(サブディレクトリ)上で 次のように入力します。

DMS6200 ↵

入力後、タイトル画面が表示されます。
この状態からDOSに戻るためには CTRL + C キーを入力します。

```
    *** E0C6200 Development tool Management System. --- Ver 1.0 ***

 EEEEEEEEEE   PPPPPPPP        SSSSSSS       00000000      NNN      NNN
 EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    000     000    NNNN     NNN
 EEE          PPP     PPP   SSS     SSS   000       000  NNNNN    NNN
 EEE          PPP     PPP   SSS           000       000  NNNNNN   NNN
 EEEEEEEEEE   PPPPPPPPPP     SSSSSS       000       000  NNN NNN NNN
 EEEEEEEEEE   PPPPPPPP          SSSS      000       000  NNN  NNNNNN
 EEE          PPP                SSS     000       000  NNN   NNNNN
 EEE          PPP             SSS   SSS   000       000  NNN    NNNN
 EEEEEEEEEE   PPP             SSSS  SSS    000     000   NNN     NNN
 EEEEEEEEEE   PPP               SSSSSSS      00000000     NNN      NN

            (C) Copyright 1991 SEIKO EPSON CORP.


                              STRIKE ANY KEY.
```

タイトル画面

メニュー画面

```
      DMS6200 Version 1.0      Copyright(C) SEIKO EPSON CORP. 1991.

1) ASM62XX .EXE
2) FOG62XX .EXE
3) ICS62XXB.BAT
4) ICS62XXW.EXE
5) MDC62XX .EXE
6) MLA628X .EXE
7) SOG62XX .EXE

Input Number ? [   ]
```

この状態からDOSに戻るためには ESC キーを入力します。

(2) タイトル画面で任意のキーを入力すると、メニュー画面が表示されます。
このメニュー画面では、拡張子が"ＥＸＥ"、"COM"、"BAT"の実行可能ファイルがすべて表示されます。S5U1C62xxxD以外の実行ファイルを実行用ディスク(サブディレクトリ)にコピーした場合は、それらのファイル名も表示されます。

メニュー選択

```
Input Number ? [1  ]
```

(3) ここで起動したい開発支援ツールの番号を入力し、リターンキーを押してください。ソースファイル選択画面が表示されます。

ソースファイル選択画面

```
      DMS6200 Version 1.0      Copyright(C) SEIKO EPSON CORP. 1991.

 1) C2XXYYY .DAT
 2) C28XYYY .MDT
 3) C28XYYY .MPR
 4) C2XXYYY .PRN
 5) C2XXYYY .SEG
 6) C28XYYYA.DOC
 7) C28XYYYA.HEX
 8) C2XXYYYF.DOC
 9) C2XXYYYF.HEX
10) C2XXYYYH.HEX
11) C2XXYYYL.HEX
12) C2XXYYYS.DOC
13) C2XXYYYS.HEX
14) C62XXYYY.PA0

 Input Number ? [1  ]

     Edit > [ASM62XX C2XXYYY                                    ]
```

(4) 表示例はASM62XXを選択した場合です。
ここでソースファイルの番号を入力します。
ESCキーを入力すると前の画面に戻ります。
ソースファイルを番号で選択すると[ ]で囲まれたエディットラインが現われますので必要に応じてオプションパラメータを入力します。エディットライン上ではBSキーが有効です。
入力が終了したらリターンキーを押します。

以上の操作でASM62XXが起動します。(MLA628Xも同様の操作で起動します。)
ソースファイルが別のドライブまたはディレクトリにある場合、メニューの中に表示されません。この場合は番号の入力をリターンキーのみでスキップし、エディットライン内にドライブ/ディレクトリとソースファイル名を入力してください。

起動時、特にソースファイルを必要としないもの(ASM62XX、MLA628X以外)については、リターンキーを2度押すことにより起動します。

オプションパラメータおよび起動後の操作については、各支援ツールの説明を参照してください。

(5) 開発支援ツールの実行が終了すると"Input Any Key"のメッセージが出力されますので、任意のキーを押してください。メニュー画面に戻ります。

```
Input Any Key ...
```

III

Cross Assembler

ASM62XX

ここでは4-bit Single Chip Microcomputer S1C62XXX用
クロスアセンブラASM62XXの操作方法と、ソース
ファイルの作成を中心に解説を行っています。

Cross Assembler ASM62XX

- 目 次 -

1 機種による相違点 III-1

2 ASM62XXの概要 III-2

2.1 概要と実行フロー III-2

2.2 ASM62XXの入出力ファイル III-2

3 ASM62XXの操作方法 III-3

3.1 ASM62XXの起動 III-3

3.2 オートページセット機能の選択 III-5

3.3 クロスリファレンステーブルの作成 III-5

4 ソースファイルの形式 III-6

4.1 ソースファイル名 III-6

4.2 ステートメント (行) III-6

4.2.1 ラベルフィールド III-6

4.2.2 ニーモニックフィールド III-6

4.2.3 オペランドフィールド III-7

4.2.4 コメントフィールド III-7

4.3 インデックス III-7

4.3.1 ラベル III-7

4.3.2 シンボル III-7

4.4 定数と演算式 III-8

4.4.1 数値定数 III-8

4.4.2 文字定数 III-8

4.4.3 演算子 III-8

4.4.4 ロケーションカウンタ III-9

4.5 疑似命令 III-10

4.5.1 データ定義擬似命令 III-10

4.5.2 メモリ設定擬似命令 III-11

4.5.3 アセンブラ制御擬似命令 III-13

4.6 マクロ機能 III-14

4.6.1 マクロ命令 III-14

4.6.2 マクロ定義 III-15

4.6.3 マクロコール III-15

5 エラーメッセージ III-18

Appendix ASM62XX実行例 III-19

ソースファイル (C2XX0A0.DAT) III-19

アセンブラの実行 (コンソールの表示) III-20

アセンブリリストファイル (C2XX0A0.PRN) III-21

オブジェクトファイル (C2XX0A0H.HEX、C2XX0A0L.HEX) III-22

1 機種による相違点

プログラムを作成するうえでは、S1C62 Familyの各機種によりメモリ容量が異なりますので、以下の点に注意する必要があります。
各機種の制限事項は"S5U1C62xxxD Manual"に記載されています。

■ ROM領域

機種によりROM容量が異なります。
このROM容量によりバンク数(16ページ/1バンク)とページ数(256ステップ/1ページ)が決定され、以下のメモリ設定擬似命令および"PSET"命令がその範囲内に制限されます。

	有効指定範囲
ORG擬似命令:	0000H～ROMの最終ステップ
PAGE擬似命令:	00H～ページ数 - 1
BANK擬似命令:	1バンク構成の機種→0Hのみ 2バンク構成の機種→0Hおよび1H
PSET命令:	00H～ページ数 - 1

ASM62XXにおいて、この有効指定範囲を越える指定が行われた場合はエラーになります。

■ RAM領域

機種によりRAM容量が異なります。
このRAM容量によりページ数(256ワード/1ページ)が決定されます。また、0番地からRAMの最終アドレスまでの間に未定義領域も含まれます。

インデックスレジスタに未定義領域のアドレスを設定した場合、そのメモリアクセスは無効となりますが、ASM62XXでエラーとはなりませんので注意してください。

■ 未定義コード

S1C62 Familyの中では機種による命令セットの違いはありません。ただし、SLP命令や、RAM容量の関係でインデックスレジスタのページ部(XP、YP)をアクセスする命令などが使用できない場合があります。

2 ASM62XXの概要

2.1 概要と実行フロー

Cross Assembler ASM62XX(以後は単にASM62XXと記述します)は4-bit Single Chip Microcomputer S1C62XXXの命令コードを出力するアセンブラで、PC-DOSのもとで使用できます。

ASM62XXはエディタ等によって作成したプログラムソースファイルをアセンブルし、Intel-HEXフォーマットのオブジェクトファイルおよびアセンブリリストファイルを出力します。
本アセンブラはマクロ定義機能によるプログラムのモジュール化、オートページセット機能によるROMのページ構造を意識しないプログラミングが可能となっています。また、プログラム容量(ROM容量)のオーバーフローや未定義コードなどの細かなエラーチェック、アセンブリリストファイルへのラベルテーブル、クロスリファレンステーブルの付加など、デバッグについても配慮されています。

アセンブラのプログラム名は"ASM62XX.EXE"です。
図2.1.1にASM62XXの実行フローを示します。

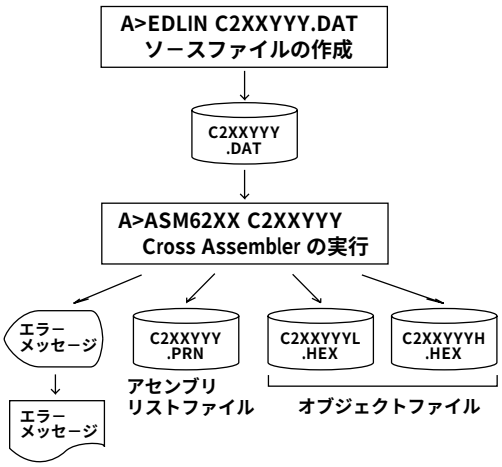


図2.1.1 ASM62XXの実行フロー

2.2 ASM62XXの入出力ファイル

ASM62XXはソースファイルを入力し、 アセンブル後にオブジェクトファイルとアセンブリリストファイルを出力します。

■ ソースファイル (C2XXYYY.DAT)

EDLINなどのエディタで作成したソースプログラムファイルです。
ファイル名は"C2XXYYY"の形式で7文字以内で作成してください。"YYY"はセイコーエプソンから連絡する機種名を参考にして付けてください。
また、拡張子は必ず".DAT"にしてください。

■ オブジェクトファイル (C2XXYYYH.HEX、C2XXYYYL.HEX)

アセンブルにより作成されるインテルヘキサフォーマットのプログラムファイルです。
S1C62XXXの命令コードが12ビットのため、上位バイト(9ビット目以上の4ビットに"0000B"の上位ビットを付加)を"C2XXYYYH.HEX"、下位バイト(8ビット以下)を"C2XXYYYL.HEX"の２つのファイルに分けて出力します。

■ アセンブリリストファイル (C2XXYYY.PRN)

ソースプログラムの各行に命令コード、エラーメッセージ(エラーがある場合)を加えたプログラムリストファイルです。ファイルの最後にはラベルテーブルとオプションによりクロスリファレンステーブルが作成されます。ファイル名は"C2XXYYY.PRN"になります。

各ファイルの内容については"Appendix"を参照してください。

3 ASM62XXの操作方法

ここでは、ASM62XXの操作方法について説明します。

3.1 ASM62XXの起動

ASM62XXを起動させるにはDOSのコマンドレベル (A>などのプロンプトが表示されている状態)において次の形式で入力を行います。
DMS6200から起動させる場合は、メニュー画面で"ASM62XX.EXE"およびソースファイルを選択し、必要に応じてオプションの入力も行います。

ASM62XX_ [ドライブ名:]ソースファイル名[.shp] _ [–H]↵

_ はスペースの入力を表します。
[ ] は省略可能なパラメータであることを表します。
↵はリターンキー(Enter key)の入力を表します。

■ ドライブ名

ソースファイルがASM62XX.EXEと異なるドライブに存在する場合は、ソースファイル名の前にそのドライブ名を入力します。
同じドライブにある場合は省略できます。

■ ソースファイル名

ASM62XXに入力するソースファイル名を入力します。

注: ソースファイル名は7文字以内で、拡張子(".DAT")は入力しないでください。

■ .shp

s、h、pはそれぞれ以下のようにファイルの入出力ドライブを指定するオプションで、省略可能です。

s: "A"から"P"でソースファイルを入力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)上のソースファイルが入力されます。
ソースファイル名の前にドライブ名が入力されている場合でも、sの指定が有効となります。

h: "A"から"P"でオブジェクトファイル(HEX)を出力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)に出力されます。
"Z"を指定した場合はアセンブルのみが行われ、オブジェクトファイルは作成されません。

p: "A"から"P"でアセンブリリストファイルを出力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)に出力されます。
"X"を指定した場合はエラーメッセージを含むリストがコンソール上に出力されます。
"Z"を指定した場合はアセンブリリストファイルは作成されません。

s、h、pは3個を同時に指定してください。1個のみの指定は行えません。

■ -Nオプション

プログラムメモリの未定義領域のコード生成(FFHのデータフィル)を行いません。

注: 提出するプログラムデータは、"-N"オプションを使用せず、未定義プログラム領域にFFHのデータを挿入してください。

例1: 基本的なアセンブル例

A>ASM62XX C2XXYYY↵	この例ではソースファイルC2XXYYY.DATがAドライブから入力され、オブジェクトファイルC2XXYYYH.HEX、C2XXYYYL.HEXとアセンブリリストファイルC2XXYYY.PRNがAドライブに作成されます。
A>ASM62XX B:C2XXYYY↵	この例ではソースファイルC2XXYYY.DATがBドライブから入力され、オブジェクトファイルC2XXYYYH.HEX、C2XXYYYL.HEXとアセンブリリストファイルC2XXYYY.PRNがBドライブに作成されます。
A>ASM62XX C2XXYYY.BBZ↵	この例ではソースファイルC2XXYYY.DATがBドライブから入力され、オブジェクトファイルC2XXYYYH.HEX、C2XXYYYL.HEXがBドライブに作成されます。アセンブリリストファイルは作成されません。

例2: -Nオプションの使用例

A>ASM62XX C2XXYYY -N↵	この例では作成されるオブジェクトファイルC2XXYYYH.HEX、C2XXYYYL.HEXにおいて未定義プログラム領域は生成されません。(Appendix ASM62XX実行例を参照してください。)
A>ASM62XX C2XXYYY↵	この場合、オブジェクトファイルの未定義プログラム領域には"FFH"のデータを挿入します。

ASM62XXが起動すると、以下の起動メッセージが表示されます。

例: C2XX0A0.DATのアセンブルを行う場合

```
A>ASM62XX C2XX0A0
             *** E0C62XX CROSS ASSEMBLER. --- Ver 2.00 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO     NNN     NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN    NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN   NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN  NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP           SSS     SSS   OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO     NNN      NN

               (C) COPYRIGHT 1991 SEIKO EPSON CORP.

             SOURCE FILE NAME IS " C2XXYYY.DAT "

             THIS SOFTWARE MAKES NEXT FILES.

                C2XXYYYH.HEX  ...  HIGH BYTE OBJECT FILE.
                C2XXYYYL.HEX  ...  LOW BYTE OBJECT FILE.
                C2XXYYY .PRN  ...  ASSEMBLY LIST FILE.
```

3.2 オートページセット機能の選択

起動メッセージの次に画面にはオートページセット機能を選択するメッセージが表示されます。

```
DO YOU NEED AUTO PAGE SET?(Y/N)
```

ここで、オートページセットを行う場合はYキー、行わない場合はNキーを入力します。
また、CTRL + Cの入力でDOSのコマンドレベルに戻すこともできます。

■ オートページセット機能

プログラムを"JP"などの分岐命令を使用して他のページに分岐させる場合、分岐命令の前に"PSET"命令による分岐先ページの設定を行う必要があります。
オートページセット機能はこの"PSET"命令の挿入を自動的に行うもので、分岐命令と分岐先のページが同一か判断し、異なる場合に"PSET"命令を挿入します。同一ページの場合は何も行いません。
したがって、ソースファイル内に"PSET"命令が正しく記述してある場合はオートページセットを行わないでください。

注: オートページセット機能を選択する場合、ソースプログラム作成に関して制限事項があります。

詳しくは、"4.3.1 ラベル"の説明を参照してください。

3.3 クロスリファレンステーブルの作成

オートページセット機能の選択後、画面にはクロスリファレンステーブルの作成を選択するメッセージが表示されます。

```
DO YOU NEED CROSS REFERENCE TABLE?(Y/N)
```

クロスリファレンステーブルを作成する場合はYキー、作成しない場合はNキーを入力します。また、CTRL + Cの入力でDOSのコマンドレベルに戻すこともできます。

注: ASM62XX起動時にアセンブリリストファイルの出力先(pオプション)を"Z"(作成しない)に指定した場合は上記のメッセージは出力されず、クロスリファレンステーブルも作成されません。

■ クロスリファレンステーブル

クロスリファレンステーブルはソースファイル内で設定されたシンボルとその使用場所を示す一覧表で、次の形式で作成されアセンブリリストファイルの最後に付け加えられます。

```
 CROSS REFERENCE TABLE   PAGE X-   1
LABEL1  4#       29        36          ....
LABEL2  15#      40
  :     :        :
```

シンボル　記述されている行番号("#"はシンボルが設定された行番号)

このテーブルはデバッグ時に参照するもので、シンボル定義の重複などはすぐに発見できます。

4 ソースファイルの形式

ソースファイルの内容はS1C62XXXのインストラクション(ニーモニック表記)と以下に説明する擬似命令によるソースプログラムで、EDLINなどのエディタで作成します。
インストラクションセットについては"S1C6200/6200AコアCPUマニュアル"、"S1C62xxxテクニカルマニュアル"を参照してください。

4.1 ソースファイル名

ソースファイルには7文字以下の任意の名前を付けることができますが、基本的には次の書式にそって付けてください。

C2XXYYY.DAT

"YYY"は3文字以内の英数字で、セイコーエプソンから連絡する機種名を参考にしてください。
また、拡張子は必ず".DAT"にします。

4.2 ステートメント (行)

ソースプログラムの各ステートメント(行)は次の書式で記述します。

基本フォーマット	**<名前>[:]**	**<命令>**	**<式>[,<式>･･･]**	**<;コメント>**
例:	ON	EQU	1	
		ORG	100H	
	START:	JP	INIT	;To Initialize
		:		
	ラベルフィールド	ニーモニックフィールド	オペランドフィールド	コメントフィールド

ステートメントはラベルフィールド、ニーモニックフィールド、オペランドフィールド、コメントフィールドの4つのフィールドで構成され、最大132文字まで記述できます。
各フィールド間には1個以上のスペース、またはタブを挿入して区切りとします。

なお、ラベルフィールド、コメントフィールドについては必要に応じて記述します。
CR(キャリッジリターン)コードのみの空白行も許されます。

ステートメント、各フィールドとも任意のカラムから開始できます(ラベルフィールドは除く)が、フィールドの区切りにタブを使用し、各フィールドの頭を揃えておくとプログラムが見やすくなります。

4.2.1 ラベルフィールド

ラベルフィールドにはメモリアドレスを参照するラベル、定数を定義するシンボル、マクロ名を記述します。ただし、名前が必要ないステートメントについては省略できます。
ラベルフィールドは第1カラム目から記述する必要があり、記述する名前には以下の制限があります。

- 長さは14文字以内です。
- ニーモニックやレジスタ名と同じ名前は使用できません。
- 以下の英数字が使用できますが、先頭の文字は必ず数字以外にします。
  A～Z、a～z、0～9、_、?
- 英小文字と大文字は区別されません。
- "??nnnn"(nは数字)という名前は使用できません。

なお、ニーモニックフィールドとの区切りにコロン":"が使用でき、この場合にはその後にスペースやタブは必要ありません。また、ラベルフィールドのみのステートメントも許されます。

4.2.2 ニーモニックフィールド

ニーモニックまたは擬似命令を記述します。

4.2.3 オペランドフィールド

各命令のオペランドを記述します。
オペランドの内容と個数は各命令のフォーマットにしたがいますが、値を指定する式として記述する内容は以下のいずれかになります。

・数値定数、文字定数、あるいは定数を定義したシンボル
・メモリアドレスを示すラベル
・指定する値を得る演算式

なお、オペランドが2つ以上の式で構成される場合はそれぞれの式をカンマ","で区切ります。

4.2.4 コメントフィールド

コメントフィールドはプログラムの見出しや処理内容等、任意のコメントを記述するもので、アセンブルの処理やアセンブルにより作成されるオブジェクトファイルには影響を与えません。

セミコロン";"から行末のCRコードまでがコメントフィールドとして見なされます。
コメントフィールドのみの行も許されますが、コメントが複数行にわたる場合でも各行のコメントの先頭にはセミコロンが必要です。

4.3 インデックス

ASM62XXでは必要な値にインデックスを付けることにより、以後はインデックスでその値を参照できます。
なお、インデックスの記述に関する制限については"4.2.1 ラベルフィールド"の説明を参照してください。

4.3.1 ラベル

ラベルはプログラムのロケーションを参照するために付けるインデックスで、インストラクションセットの中で、メモリアドレスを即値データとして指定するオペランドに使用できます。たとえば、分岐先のステートメントにラベルを記述しておくことで、"JP"命令などのオペランドにそのラベルが使用できます。
EQU命令、またはSET命令のラベルフィールドに付けられた名前はラベルではなくシンボルと見なされます。

例:

```
                    JP      NZ,LABEL1
                    :
LABEL1:             LD      A,0
```

任意のステートメントにラベルを付けることができますが、以下の擬似命令に付けられたラベルは無視されます。

ORG、BANK、PAGE、SECTION、END、LABEL、ENDM

注: オートページセット機能(3.2項参照)を選択する場合は"JP"命令と"CALL"命令の1つ前の行にラベルだけの行を記述する必要があります。

例:

```
PGSET:
                    JP      LABEL
```

4.3.2 シンボル

シンボルは数値定数や文字定数を示すインデックスで、その値を参照する前に(一般的にはプログラムの最初に)定義しておきます。定義したシンボルはインストラクションセットの中で即値データを指定するオペランドに使用できます。

例:

```
ON          EQU     1       ("EQU"については4.4.1項参照)
OFF         EQU     0
              :
            LD      A,ON        ; = LD A,1
              :
            LD      A,OFF       ; = LD A,0
              :
```

4.4 定数と演算式

以下に即値データを指定する場合の表記方法などについて説明します。

4.4.1 数値定数

数値定数はASM62XX内では13ビットの値として扱われ、それ以上の数値を記述した場合14ビット目以降は無視されます。
なお、実際の有効桁数は各命令のオペランドにより異なりますので注意してください。オペランドの有効桁数を越えた場合はエラーになります。

```
例:     ABC  EQU  0FFFFH     →   ABCは1FFFHに定義される
        LD   A,65535    →   有効桁数(4ビット)を越えているためエラー
```

基数はデフォルトで10進数として見なされますが、基数を指定する場合は以下のように記述します。

2進数:	数値の後に"B"を付けます。	**例:** `1010B (=10), 01100100B (=100)`
8進数:	数値の後に"O"または"Q"を付けます。	**例:** `012O (=10), 144Q (=100)`
10進数:	数値のみまたは後に"D"を付けます。	**例:** `10, 100D (=100)`
16進数:	数値の後ろに"H"を付けます。 先頭が"A"～"F"で始まる場合、名前と区別するためにその前に"0"を付けます。	**例:** `0AH (=10), 64H (=100)`

4.4.2 文字定数

文字定数はASCIIキャラクタを表すもので、アポストロフィ(' ')で囲んで記述します。
1文字のASCIIキャラクタを記述した場合、8ビットのデータとして扱われ、2文字以上の記述を行った場合は最後の2文字のみが13ビットのデータとして有効となります。

```
例:     'A' (=41H), 'BC' (=0243H), 'PQ' (=1051H)
        'DEFGH' → 'GH' (=0748H) 'DEF'は無視
```

アポストロフィ自体は文字定数として扱うことはできませんので、"27H"や"39"などの数値定数として記述してください。

4.4.3 演算子

オペランドなどの値を指定する場合、定数の代わりに演算式を記述し、その演算結果を値として用いることができます。
式の項として定数の他にラベルやシンボルなども使用できますが、これらの値は13ビットデータ(14ビット目以上は無視)として演算が行われ、演算結果も13ビットで得られます。
ただし、演算結果が各命令によるオペランドの有効桁数を越えた場合はエラーになります。
演算子は算術演算子、論理演算子、関係演算子の3種類に分けられ、それぞれ以下のとおりです。a、bは項を、_は1個以上のスペースを表します。

■ 算術演算子

算術演算子は加減乗除に加え、ビットのシフトや分離を含めた以下の11種類です。

+a : 単項マイナス(続く値が正であることを示す)
-a : 単項プラス (続く値が負であることを示す)
a+b : 加算(符号なし)
a-b : 減算(符号なし)
a*b : 乗算(符号なし)
a/b : 除算(符号なし)
a_MOD_b : a/bの剰余
a_SHL_b : aをbビット分左シフト ← □□□□□□□□ ← 0
例: `00000011B SHL 2 → 00001100B`
a_SHR_b : aをbビット分右シフト 0 → □□□□□□□□ →
例: `11000011B SHR 2 → 00110000B`
HIGH_a : a(13ビット)の上位8ビットを分離 **例:** `HIGH 1234H → 12H`
LOW_a : a(13ビット)の下位8ビットを分離 **例:** `LOW 1234H → 34H`

■ 論理演算子

論理演算子は以下の4種類で、それぞれの項を論理演算した結果を返します。

a_AND_b : 論理積　　**例:** 00001111B AND 00000011B → 00000011B
a_OR_b : 論理和　　**例:** 00001111B OR 11110000B → 11111111B
a_XOR_b : 排他的論理和　　**例:** 00001111B XOR 00000011B → 00001100B
NOT_a : 論理否定　　**例:** NOT 00001111B → 11110000B

■ 関係演算子

関係演算子は2つの項を比較し、演算子どおりの関係にある場合に真(1FFFH)、それ以外の場合に偽(0)が返されます。

a_EQ_b : aとbが等しい場合に真
a_NE_b : aとbが等しくない場合に真
a_LT_b : aがbより小さい場合に真
a_LE_b : aがbより小さいか等しい場合に真
a_GT_b : aがbより大きい場合に真
a_GE_b : aがbより大きいか等しい場合に真

以上の演算子で項との間に_が表記されているものについては、必ずスペースを挿入してください。
また、演算子はすべて大文字で入力します。

式には複数の演算子やカッコ( )を含めることができます。この場合、基本的には式の左から評価されますが、カッコ( )や演算子の優先順位によっては優先順位の高い演算から行われます。また、左右のカッコは必ず同数の対になっている必要があります。

以下に演算子の優先順位を示します。

演　算　子	優先順位
)	低い
OR, XOR	↑
AND	
EQ, NE, LT, LE, GT, GE	
+(加算), -(減算)	
*, /, MOD, SHL, SHR	
(	↓
HIGH, LOW, NOT, -(単項マイナス), +(単項プラス)	高い

例: 以下は演算式の例です(ABC = 1、BCD = 3)。

```
LD A,BCD*(ABC+1)  ; A <- 6

LD A,ABC LT BCD   ; A <- 0FH (1111B)

OR B,ABC SHL BCD
; Set bit3 in B-register (= OR B,1000H)

AND B,ABC SHL BCD XOR 0FH
; Reset bit3 in B-register (= AND B,0111B)
```

4.4.4 ロケーションカウンタ

各命令コードの開始アドレスは各ステートメントのアセンブル時にロケーションカウンタにセットされます。このロケーションカウンタの値をプログラムで使用するには、ラベルのほかに"$"を用いる方法があります。

■ ロケーションカウンタ

バンク(1ビット)、ページカウンタ(4ビット)、ステップカウンタ(8ビット)の13ビットで構成されます。

	バンク	ページカウンタ				ステップカウンタ							
BIT	12	11	10	9	8	7	6	5	4	3	2	1	0
内容	バンク BNK	ページアドレス PCP				ステップアドレス PCS							

例: ロケーションカウンタ

```
        (BNK)     (PCP)     (PCS)
          0         1         02          JP          $+3
```

この例ではロケーションカウンタは"JP"命令の開始アドレスを示し、"$"にはPCSの値(02)が代入されます。したがって、"JP5"としてアセンブルが行われ、プログラム実行時は3ステップ先(PCS=05)にジャンプします。

4.5 擬似命令

擬似命令はデータ定義、メモリ設定、アセンブラ制御、マクロ機能の4種類に分けられます。
前述の演算式なども含め、これらの擬似命令はアセンブラに対するもので、開発後のプログラム上で実行されるものではありません。

以下の説明の中で、擬似命令のフォーマットに使用されている< >はステートメント中に記述する内容を表します(< >自体は記述しません)。_は1個以上のスペース、またはタブを表します。また、<式>にはシンボル、定数、演算式などを記述します。
(マクロ機能については4.6項で説明します。)

4.5.1 データ定義擬似命令

データ定義擬似命令にはシンボルを定義するEQU命令とSET命令、プログラムメモリにデータをセットするDW命令があります。

■ EQU (Equation)

<シンボル>_EQU_<式>	シンボル定義

EQU命令はラベルフィールドに記述した<シンボル>にオペランドフィールドの<式>の値を定義します。
<式>に13ビットを越える値を指定した場合、14ビット目以上は無視されます。
この定義はシンボルがプログラム内で参照される前に行う必要があります。未定義のシンボルを参照した場合、Uエラーになります。また、すでに定義されているシンボルを再定義することはできません。再定義を行った場合、Pエラーになります。

例:
```
ZERO    EQU     30H
ONE     EQU     ZERO+1
ONE     EQU     31H             ← ONEが２重定義のため、Pエラー
FOUR    EQU     TWO*2           ← TWOが未定義のため、Uエラー
```

■ SET

<シンボル>_SET_<式>	シンボル定義

SET命令もEQU命令と同様に<シンボル>に<式>の値を定義します。
SET命令ではシンボルの再定義が可能です。

例:
```
BIT  SET  1
 :
BIT  SET  2                     ← 再定義が可能
 :
BIT  SET  BIT SHL 1             ← 直前までの定義内容が参照可能
```

■ DW (Define Word)

<ラベル>_DW_<式>	データのセット

DW命令はロケーションカウンタで示される現在のアドレスに<式>の内容(12ビットを越えるものについては下位12ビット)を割り付け、オブジェクトコードとします。

例: ロケーションカウンタ
```
(BNK)    (PCP)    (PCS)
  0        2        0A      TABLE     DW 141H ; = RETD 'A'
  0        2        0B                DW 142H ; = RETD 'B'
  0        2        0C                DW 143H ; = RETD 'C'
 :
```
<ラベル>は省略可能です。

4.5.2 メモリ設定擬似命令

S1C62XXXのプログラムメモリは256ステップ単位にページ分けされていますが、プログラム作成時のメモリ管理はソースプログラム上で行う必要があります。具体的にはプログラムのロケーションの設定やページ境界の設定がこれにあたります。
メモリ設定擬似命令はこれらの指定を行うもので、アセンブラはこの指定をもとにロケーションカウンタを設定します。
すでに使用されているメモリ領域を指定したり、指定が行われずにページを越えるステートメントがある場合、アセンブラは"!"(Warning)のみを表示し、次に正しい指定が行われるまでのステートメントを無効としてアセンブルを行いますので注意してください。
また、オートページセット機能を使用する場合は"PSET"命令が挿入される余裕を各ページに作っておく必要があります。

■ ORG (Origin)

ORG_<式>　　ロケーションカウンタの設定

ORG命令は<式>の値をロケーションカウンタに設定します。
プログラムの先頭にORG命令がない場合、ロケーションカウンタは0(BNK=0、PCP=0、PCS=0)に設定されアセンブルが行われます。
ORG命令はプログラム内の複数箇所で使用することができます。ただし、現在のロケーションカウンタより前のロケーションに設定することはできません。このような指定を行った場合には次に正しい設定が行われるまでのステートメントがすべて無効となり、"!"(Warning)が表示されます。
ORG命令の前にラベルを記述することもできますが、ラベルテーブルには登録されないため、参照はできません。この場合はORG命令の次のステートメントにラベルを記述してください。

例:
```
        ORG  0100H     ; BNK=0, PCP=1, PCS=00H
START   :
```

ROM容量を越えた値の指定を行った場合、Rエラーになります。

注: 各機種によりプログラムメモリの上限は異なります。
("S5U1C62xxxD Manual"を参照してください。)

■ BANK

BANK_<式>　　バンク(BNK)の設定

BANK命令は<式>の値をバンク(BNK)に設定し、ページカウンタ(PCP)とステップカウンタ(PCS)を00Hにします。
BANK命令はプログラム内の任意の箇所で使用することができます。
ただし、現在のバンク(00ページ、00ステップでの指定を除く)またはそれより前のバンクに設定することはできません。このような指定を行った場合には"!"(Warning)が表示され、次に正しい設定が行われるまでのステートメントがすべて無効となります。
BANK命令の前にラベルを記述することもできますが、ラベルテーブルには登録されないため、参照はできません。この場合はBANK命令の次のステートメントにラベルを記述してください。

■ PAGE

PAGE_<式>　　ページカウンタ(PCP)の設定

PAGE命令は<式>の値をページカウンタ(PCP)に設定し、ステップカウンタ(PCS)を00Hにします。
PAGE命令はプログラム内の複数箇所で使用することができます。ただし、現在のページ(00ステップでの指定を除く)またはそれより前のページに設定することはできません。このような指定を行った場合には"!"(Warning)が表示され、次に正しい設定が行われるまでのステートメントがすべて無効となります。PAGE命令の前にラベルを記述することもできますが、ラベルテーブルには登録されないため、参照はできません。この場合はPAGE命令の次のステートメントにラベルを記述してください。

例: ロケーションカウンタ

```
(BNK)  (PCP)  (PCS)
  :      :      :            :      :
  0      0     1AH           LD     X,0
  0      0     1BH           LD     Y,0
  :      :      :            :      :
  0      0     F0H           JP     xxx

                             PAGE   2
  0      2     00H   SUB1:   LD     A,MX
  0      2     01H           LD     B,MY
  :      :      :            :      :

                             PAGE   1
!                    SUB2:   LD     A,MX
!                            LD     B,MY
                             :      :
                             PAGE   3
  0      3     00H   SUB3:   LD     A,0
  0      3     01H           LD     B,1
  :      :      :            :      :
```

前のページを指定したため無効

有効

最終ページを越えた値の指定を行った場合、Rエラーになります。

注: 各機種により最終ページは異なります。
("S5U1C62xxxD Manual"を参照してください。)

■ SECTION

SECTION	セクションの変更

SECTION命令は次のセクションの先頭アドレスをロケーションカウンタに設定します。
セクションとはプログラムメモリの先頭から16ステップ単位に区切られた領域をいいます。

```
(BNK)  (PCP)  (PCS)
  0      1     00H
  0      1     10H
  0      1     20H
  :      :      :
  0      1     F0H
  0      2     00H
  0      2     20H
  :      :      :
  0      3     F0H
```

セクション	1
セクション	2

16ステップ

セクション	16
セクション	17

セクション	48

ページの最終セクションにあるSECTION命令は、ステップカウンタを00Hにすると同時にページカウンタも更新しますので、新たにページを設定する必要はありません。
SECTION命令の前にラベルを記述することもできますが、ラベルテーブルには登録されないため、参照はできません。この場合はSECTION命令の次のステートメントにラベルを記述してください。

例: ロケーションカウンタ

```
(BNK)  (PCP)  (PCS)
  :      :      :            :          :
  0      1     09H           JPBA
  0      1     0AH           LD         X,0
  0      1     0BH           LD         Y,0
  0      1     0CH           LD         MX,4

                             SECTION
  0      1     10H    TABLE  LD         A,1
  0      1     11H           ADD        A,1
  :      :      :            :          :
  0      1     FAH           RET

                             SECTION
  0      2     00H    LOOP   SCF
  0      2     01H           ADD        A,MY
  :      :      :            :          :
```

4.5.3 アセンブラ制御擬似命令

■ END

END	アセンブル終了

アセンブラはEND命令があるステートメントでアセンブルを終了します。
END命令以降にあるステートメントはすべて無効となります。
この命令がない場合、アセンブラが終了しないことがありますので、必ず最後に記述してください。
END命令の前にラベルを記述することもできますが、ラベルテーブルには登録されないため、参照はできません。

4.6 マクロ機能

プログラムで同じステートメントブロックを複数箇所で使用する場合、そのステートメントブロックをあらかじめ任意の名前で定義しておくことにより、以後はその名前で呼び出すことができます。
この定義されたステートメントブロックをマクロと呼びます。
マクロはサブルーチンとは異なり、呼び出しを行った場所のすべてにステートメントブロックが展開されますので、ステートメントブロックの大きさや使用頻度によりサブルーチンと使い分けてください。

4.6.1 マクロ命令

ASM62XXにはページ間の分岐を"PSET"命令によるページ指定なしに行えるように、あらかじめ以下の命令が定義されています。これらの命令をマクロ命令と呼びます。

マクロ命令	展開後のニーモニック	コード 11	10	9	8	7	6	5	4	3	2	1	0
JPM ps	PSET p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	JP s	0	0	0	0	s7	s6	s5	s4	s3	s2	s1	s0
JPM C,ps	PSET p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	JP C,s	0	0	1	0	s7	s6	s5	s4	s3	s2	s1	s0
JPM NC,ps	PSWT p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	JP NC,s	0	0	1	1	s7	s6	s5	s4	s3	s2	s1	s0
JPM Z,ps	PSET p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	JP Z,s	0	1	1	0	s7	s6	s5	s4	s3	s2	s1	s0
JPM NZ,ps	PSET p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	JP NZ,s	0	1	1	1	s7	s6	s5	s4	s3	s2	s1	s0
CALLM ps	PSET p	1	1	1	0	0	1	0	p4	p3	p2	p1	p0
	CALL s	0	1	0	0	s7	s6	s5	s4	s3	s2	s1	s0

psは分岐先のアドレスを示す13ビットの即値データで、ラベルが使用できます。

例: ソースファイル

```
            :
            JPM       LABEL2
            :
            PAGE      2
LABEL2      LD        A,0
            :
```

展開後（アセンブリリストファイル）

```
            :
            JPM       LABEL2
+            PSET     LABEL2
+            JP       LABEL2
            :
            PAGE      2
LABEL2      LD        A,0
            :
```

4.6.2 マクロ定義

マクロ定義は次のMACRO命令、ENDM命令(擬似命令)によって定義します。

■ MACRO、ENDM

```
<マクロ名>_MACRO_ [<仮引数>, ····]
        ステートメント
            :
        ENDM
```

MACRO命令とENDM命令によって囲まれたステートメントブロックをマクロ定義します。
マクロ名はマクロを呼び出すための名前で任意に付けられますが、使用できる文字や長さはラベルフィールドの規則にしたがいます。

マクロは呼び出し時に引数を受け取ることができます。この場合、マクロ内では引数を用いるオペランドに仮引数として任意のシンボルを使用し、そのシンボルをMACROの後に記述します。複数の仮引数がある場合はそれぞれの間をカンマ","で区切ります。

マクロ定義はENDM命令によってのみ終了しますので、必ず記述してください。

例:
```
LDM   MACRO   REG,ADDR
      LD      X,ADDR
      LD      REG,MX
      ENDM
```

ADDRで指定するデータメモリからREGで指定するAまたはBレジスタにデータをロードするマクロ例です。
呼び出し例　LDM A,10H

これらの仮引数はマクロ展開時に実引数に置き換えられます。

■ LOCAL

マクロ内にラベルを持つ場合、そのマクロが複数箇所で展開されるとラベルが重複してエラーになってしまいます。これを防ぐために次のLOCAL命令があります。

```
LOCAL_<ラベル名>[, <ラベル名>····]
```

LOCAL命令によって指定されたラベルは、マクロ展開時に"?? nnnn"という名前に置き換えられます。
nnnnは4桁の10進数で、0001～9999の値を順次割り付けます。
なお、LOCAL命令はマクロ内の先頭に記述してください。他の命令が先に記述された場合LOCAL命令は無効となります。

例:
```
WAIT    MACRO   CNT
        LOCAL   LOOP
        LD      A,CNT
LOOP    SBC     A,1        ← LOOPを"?? nnnn"に置き換えて展開します。
        JP      NZ,LOOP
        ENDM
```

4.6.3 マクロコール

定義されたマクロはプログラムの任意の場所から次の書式で呼び出すことができます。

```
[<ラベル>]_<マクロ名>_[<実引数>,····]
```

マクロはマクロ名で呼び出すことができます。
引数が必要な場合はマクロ定義の仮引数に対応させて実引数を記述します。複数の実引数がある場合はそれぞれの間をカンマ","で区切ります。
実引数と仮引数は左から順に対応していきますが、実引数が仮引数より多い場合は余った実引数が無視され、逆に少ない場合は余った仮引数がヌル(00H)に置き換えられます。
マクロ名の前のラベルは任意に記述できます。

例: ソースファイル

```
                ORG     0200H

CTAS    EQU     00H
CTAE    EQU     02H
CAFSET  EQU     0101B
CAFRST  EQU     0000B
CTBS    EQU     10H
CTBE    EQU     08H
CBFSET  EQU     0001B
CBFRST  EQU     0100B

COUNT   MACRO   FSET,FRST,CTS,CTE
        LOCAL   LOOP1
        SET     F,FSET
        RST     F,FRST
        LD      A,0
        LD      X,CTS
LOOP1   ACPX    MX,A
        CP      XL,CTE
        JP      NZ,LOOP1
        ENDM

COUNTA  COUNT   CAFSET,CAFRST,CTAS,CTAE
        RET

COUNTB  COUNT   CBFSET,CBFRST,CTBS,CTBE
        RET

        END
```

アセンブル後のアセンブリリストファイルを次ページに示します。

アセンブリリストファイル

```
LISTING OF ASM62XX     C2XX0A1.PRN  ........ PAGE   1
  LINE BANK PCP PCS    OBJ          SOURCE STATEMENT
     1                                    ORG     0200H
     2
     3                0000=      CTAS     EQU     00H
     4                0002=      CTAE     EQU     02H
     5                0005=      CAFSET   EQU     0101B
     6                0000=      CAFRST   EQU     0000B
     7                0010=      CTBS     EQU     10H
     8                0008=      CTBE     EQU     08H
     9                0001=      CBFSET   EQU     0001B
    10                0004=      CBFRST   EQU     0100B
    11
    12                           COUNT    MACRO   FSET,FRST,CTS,CTE
    13                                    LOCAL   LOOP1
    14                                    SET     F,FSET
    15                                    RST     F,FRST
    16                                    LD      A,0
    17                                    LD      X,CTS
    18                           LOOP1    ACPX    MX,A
    19                                    CP      XL,CTE
    20                                    JP      NZ,LOOP1
    21                                    ENDM
    22
    23                            COUNTA  COUNT   CAFSET,CAFRST,CTAS,CTAE
    24    0   2  00    F45     +          SET     F,CAFSET
    25    0   2  01    F50     +          RST     F,CAFRST
    26    0   2  02    E00     +          LD      A,0
    27    0   2  03    B00     +          LD      X,CTAS
    28    0   2  04    F28     + ??0001   ACPX    MX,A
    29    0   2  05    A52     +          CP      XL,CTAE
    30    0   2  06    704     +          JP      NZ,??0001
    31    0   2  07    FDF                RET
    32
    33                            COUNTB  COUNT   CBFSET,CBFRST,CTBS,CTBE
    34    0   2  08    F41     +          SET     F,CBFSET
    35    0   2  09    F54     +          RST     F,CBFRST
    36    0   2  0A    E00     +          LD      A,0
    37    0   2  0B    B10     +          LD      X,CTBS
    38    0   2  0C    F28     + ??0002   ACPX    MX,A
    39    0   2  0D    A58     +          CP      XL,CTBE
    40    0   2  0E    70C     +          JP      NZ,??0002
    41    0   2  0F    FDF                RET
    42
    43                                    END
```

5 エラーメッセージ

アセンブル中にエラーが発生した場合、ASM62XXは以下のエラーシンボル、あるいはエラーメッセージをコンソールとアセンブリリストファイルに出力します。

エラーシンボルはエラーが発生したステートメントの先頭(第1カラム)に1つだけ(複数のエラーがある場合、優先順位の高いものが)出力されます。

以下のエラーシンボルは優先順位の高いものから順に示しています。

エラーメッセージ	意　　味
S (Syntax Error)	文法的に重大な誤りがある。
U (Undefined Error)	オペランドのラベル、シンボルが未定義である。
M (Missing Label)	ラベルフィールドの記述がない。
O (Operand Error)	オペランドに文法的な誤りがある。または評価できない。
P (Phase Error)	ラベル、シンボルが二重定義された。
R (Range Error)	・ロケーションカウンタの値がプログラムメモリの上限を越えた。または上限を越えるロケーションを指定した。 ・オペランドの有効桁数を越えた値が指定された。
! (Warning)	・"PAGE"命令と"ORG"命令のいずれか、または双方の指定によりメモリ領域が重複した。 ・ロケーションの指定が行われないままステートメントがページ境界を越えた。
FILE NAME ERROR	ソースファイル名が8文字以上である。
FILE NOT PRESENT	指定されたソースファイルが存在しない。
DIRECTORY FULL	指定ディスクのディレクトリに余裕がない。
FATAL DISK WRITE ERROR	ディスク上にファイルが書き込めない。
LABEL TABLE OVERFLOW	ラベル、シンボルがラベルテーブルの容量(4000個)を越えて定義された。
CROSS REFERENCE TABLE OVERFLOW	ラベル、シンボルの参照回数がクロスリファレンステーブルの容量を越えた(クロスリファレンステーブル作成時のみ)。

Appendix ASM62XX実行例

■ ソースファイル (C2XX0A0.DAT)

```
A>TYPE C2XX0A0.DAT
;
;*******<< SAMPLE PROGRAM :E0C62XX >>*******
;
ABC     EQU     0F0H
TEN     EQU     10
;
START   LD      A,0
        LD      X,8
        LD      Y,3
        LDPX    A,MX
;
        ORG     0E0H
;
NEXT    ADD     B,TEN
        LD      MX,XH
        AND     A,101B
        FAN     MY,A
        RCF
        SCPX    MX,B
        JP      C,NEXT
;
;-------<<          ERROR          >>-------
        EQU     0CH-2
ERROR   EQU     4
ERROR   LD      A,3
        SBD     MX,A
        INC     Z
        JP      UNDEF
        ORG     11100000B
        NOP5
        SECTION
        ORG     ABC+0FH
        NOP7
        NOP7
        END
```

■ アセンブラの実行 (コンソールの表示)

```
A>ASM62XX C2XX0A0
     *** E0C62XX CROSS ASSEMBLER. --- VERSION 2.00 ***

   EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO     NNN     NNN
   EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN    NNN
   EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN   NNN
   EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN  NNN
   EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
   EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
   EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
   EEE          PPP           SSS     SSS   OOO      OOO   NNN    NNNN
   EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
   EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO     NNN      NN

            (C) COPYRIGHT 1991  SEIKO EPSON CORP.

           SOURCE FILE NAME IS " C2XX0A0.DAT "

           THIS SOFTWARE MAKES NEXT FILES.

              C2XX0A0.H.HEX  ...  HIGH BYTE OBJECT FILE.
              C2XX0A0.L.HEX  ...  LOW BYTE OBJECT FILE.
              C2XX0A0 . PRN  ...  ASSEMBLY LIST FILE.


           DO YOU NEED AUTO PAGE SET?(Y/N) N
           DO YOU NEED CROSS REFERENCE TABLE?(Y/N) Y
           M   23               000A=             EQU       0CH-2
           P   24               0004=      ERROR  EQU       4
           P   25    0   0  E7    E03       ERROR  LD        A,3
           S   26    0   0  E8    FFF              SBD       MX,A
           O   27    0   0  E9    FFF              INC       Z
           U   28    0   0  EA    000              JP        UNDEF
           !   30                                  NOP5
           R   34    0   1  00                     NOP7

           8 ERROR OR WARNING(S) DETECTED
           USED : 6/2000  SYMBOLS
A>
```

■ アセンブリリストファイル (C2XX0A0.PRN)

```
A>TYPE C2XX0A0.PRN
LISTING OF ASM62XX     C2XX0A0.PRN  ........  PAGE   1
  LINE BANK PCP PCS    OBJ          SOURCE STATEMENT
     1                            ;
     2                            ;*******<< SAMPLE PROGRAM :E0C62XX >>*******
     3                            ;
     4                 00F0=      ABC     EQU     0F0H
     5                 000A=      TEN     EQU     10
     6                            ;
     7    0   0  00    E00        START   LD      A,0
     8    0   0  01    B08                LD      X,8
     9    0   0  02    803                LD      Y,3
    10    0   0  03    EE2                LDPX    A,MX
    11                            ;
    12                                     ORG     0E0H
    13                            ;
    14    0   0  E0    C1A        NEXT    ADD     B,TEN
    15    0   0  E1    EA6                LD      MX,XH
    16    0   0  E2    C85                AND     A,101B
    17    0   0  E3    F1C                FAN     MY,A
    18    0   0  E4    F5E                RCF
    19    0   0  E5    F39                SCPX    MX,B
    20    0   0  E6    2E0                JP      C,NEXT
    21                            ;
    22                            ;-------<<          ERROR          >>-------
M   23                 000A=              EQU     0CH-2
P   24                 0004=      ERROR   EQU     4
P   25    0   0  E7    E03        ERROR   LD      A,3
S   26    0   0  E8    FFF                SBD     MX,A
O   27    0   0  E9    FFF                INC     Z
U   28    0   0  EA    000                JP      UNDEF
    29                                     ORG     11100000B
!   30                                     NOP5
    31                                     SECTION
    32                                     ORG     ABC+0FH
    33    0   0  FF    FFF                NOP7
R   34    0   1  00                       NOP7
    35                                     END
8 ERROR OR WARNING(S) DETECTED
  LABEL TABLE          PAGE L-  1
  ABC     =00F0       ERROR   =0004       NEXT    0-0-E0       START    0-0-00
  TEN     =000A     U UNDEF   0-0-00
 CROSS REFERENCE TABLE   PAGE X-  1
ABC    4#       32
ERROR  24#      25#
NEXT   14#      20
START  7#
TEN    5#       14
UNDEF  28
```

■ オブジェクトファイル (C2XX0A0H.HEX, C2XX0A0L.HEX)

```
A>TYPE C2XX0A0L.HEX
:10000000000803E2FFFFFFFFFFFFFFFFFFFFFFF0F
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:10008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80
:10009000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF70
:1000A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF60
:1000B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:1000C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF40
:1000D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:1000E0001AA6851C5E39E003FFFF00FFFFFFFFF3C
:1000F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF10
:10010000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
:10011000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEF
:10012000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDF
:10013000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCF
:10014000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBF
:10015000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAF
:10016000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9F
:10017000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8F
:10018000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
:10019000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6F
:1001A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5F
:1001B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4F
:1001C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3F
:1001D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2F
:1001E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1F
:1001F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0F
:10020000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE
:10021000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEE
:10022000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDE
:10023000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCE
:10024000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBE
:10025000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAE
:10026000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9E
:10027000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8E
:10028000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7E
:10029000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6E
:1002A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5E
:1002B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4E
:1002C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3E
:1002D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2E
:1002E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1E
:1002F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0E
:10030000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD
:10031000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFED
:10032000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDD
:10033000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCD
:10034000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBD
:10035000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAD
:10036000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9D
:10037000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8D
:10038000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7D
:10039000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6D
:1003A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5D
:1003B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4D
:1003C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3D
:1003D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2D
:1003E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1D
:1003F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0D
:00000001FF
```

```
A>TYPE C2XX0A0H.HEX
:100000000E0B080EFFFFFFFFFFFFFFFFFFFFFFCD
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:10008000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF80
:10009000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF70
:1000A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF60
:1000B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:1000C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF40
:1000D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:1000E0000C0S1C0F0F0F020E0F0F00FFFFFFFFF94
:1000F000FFFFFFFFFFFFFFFFFFFFFFFFFFFF0F00
:10010000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
:10011000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEF
:10012000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDF
:10013000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCF
:10014000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBF
:10015000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAF
:10016000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9F
:10017000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8F
:10018000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7F
:10019000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6F
:1001A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5F
:1001B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4F
:1001C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3F
:1001D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2F
:1001E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1F
:1001F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0F
:10020000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE
:10021000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEE
:10022000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDE
:10023000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCE
:10024000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBE
:10025000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAE
:10026000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9E
:10027000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8E
:10028000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7E
:10029000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6E
:1002A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5E
:1002B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4E
:1002C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3E
:1002D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2E
:1002E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1E
:1002F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0E
:10030000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD
:10031000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFED
:10032000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDD
:10033000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFCD
:10034000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBD
:10035000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFAD
:10036000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9D
:10037000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF8D
:10038000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7D
:10039000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF6D
:1003A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5D
:1003B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF4D
:1003C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3D
:1003D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF2D
:1003E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1D
:1003F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF0D
:00000001FF
```

(ROM容量1,024ステップの場合)

注: オブジェクトファイルは、各機種ごとのROM容量により大きさが異なります。
ROM容量については"S5U1C62xxxD Manual"を参照してください。

IV

Melody Assembler

MLA628X

ここでは4-bit Single Chip Microcomputer S1C62N8X用
メロディアセンブラMLA628Xの操作方法と、ソース
ファイルの作成を中心に解説を行っています。

Melody Assembler MLA628X

- 目 次 -

1 機種による相違点 IV-1

2 MLA628Xの概要 IV-1

2.1 概要と実行フロー IV-1

2.2 MLA628Xの入出力ファイル IV-1

3 MLA628Xの起動方法 IV-2

4 ソースファイルの形式 IV-4

4.1 ソーファイル名 IV-4

4.2 ステートメント (行) IV-4

5 擬似命令 IV-6

6 エラーメッセージ IV-7

Appendix 入出力ファイル例 IV-8

1 機種による相違点

メロディ機能を持たない機種(S1C62N8X以外)のソフトウェアツールにはMLA628Xが含まれません。
メロディ機能を持つ機種(S1C62N8X)についても、機種によりメロディROM容量が異なります。
これにより、メロディデータ数およびそのビット構成が異なりますので注意してください。

各機種の制限事項は"S5U1C62N8xD Manual"に記載されています。

2 MLA628Xの概要

2.1 概要と実行フロー

Melody Assembler MLA628Xはエディタ等によって作成した音階データ、音符データおよびメロディオプション等で構成されたメロディのソースファイルをアセンブルし、Intel-HEXフォーマットのオブジェクトファイルおよびアセンブリリストファイル、ドキュメントファイルを出力します。

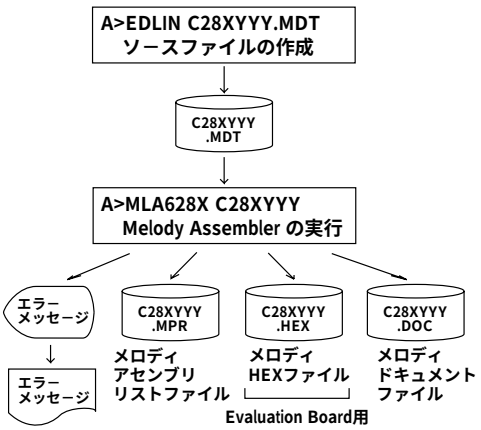


図2.1.1 Melody Assembler MLA628Xの実行フロー

2.2 MLA628Xの入出力ファイル

MLA628Xはソースファイルを入力し、アセンブル後にメロディHEXファイルとアセンブリリストファイルおよびドキュメントファイルを出力します。

■ **ソースファイル(C28XYYY.MDT)**
メロディのデータを記述します。3章を参照して、エディタで作成してください。

■ **メロディHEXファイル(C28XYYYA.HEX)**
Evaluation Board(S5U1C62N8xE)のメロディROMを作成するためのHEXファイルです。このファイルからROMを1個作成し、Evaluation Boardに搭載してください。また、ICEのOPTLDコマンドによってこのファイルを読み込み、Evaluation Boardにロードすることもできます。

注: HEXファイルをEPROMに書き込む際は、ROMの未使用領域をすべてFFHに設定してください。

ICE(S5U1C62000H)によりロードする場合は、このファイルのフォーマットがチェックされ、メロディデータの範囲を越えている場合はエラーとなります。
各機種の制限事項は"S5U1C62N8xD Manual"を参照してください。

■ **メロディアセンブリリストファイル(C28XYYY.MPR)**
ソースファイルの各行にメロディROMデータ、エラーメッセージ(エラーがある場合)を加えたメロディROMリストファイルです。ファイルの最後にはスケールROMテーブルが作成されます。

■ **メロディドキュメントファイル(C28XYYYA.DOC)**
マスクパターンを作成するためのデータファイルです。最終的に完成したこのファイルをMask Data Checker MDC628Xによって他のファイルと1つにまとめてセイコーエプソンに提出してください。

3 MLA628Xの起動方法

MLA628Xを起動させるにはDOSのコマンドレベル (A>などのプロンプトが表示されている状態)において次の形式で入力を行います。
DMS6200から起動させる場合は、メニュー画面で"MLA628X.EXE"およびソースファイルを選択し、必要に応じてオプションの入力も行います。

MLA628X_ [ドライブ名:]ソースファイル名[.shp] _ [–H]↵

_はスペースの入力を表します。
[ ]は省略可能なパラメータであることを表します。
↵はリターンキー(Enter key)の入力を表します。

ドライブ名: ソースファイルがMLA628X. EXEと異なるドライブに存在する場合はソースファイル名の前にそのドライブ名を入力します。
同じドライブにある場合は省略できます。

ソースファイル名: MLA628Xに入力するソースファイル
注: ソースファイル名は7文字以内で、拡張子(".MDT")は入力しないでください。

.shp: s、h、pはそれぞれ以下のようにファイルの入出力ドライブを指定するオプションで、大文字/小文字の入力および省略が可能です。

s…"A"から"P"でソースファイルを入力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)上のソースファイルが入力されます。
ソースファイル名の前にドライブ名が入力されている場合でも、sの指定が有効となります。

h…"A"から"P"でメロディHEXファイルとメロディドキュメントファイルを出力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)に出力されます。
"Z"を指定した場合はアセンブルのみが行われ、メロディHEXファイルおよびメロディドキュメントファイルは作成されません。

p…"A"から"P"でアセンブリリストファイルを出力するドライブを指定します。
"@"を指定した場合はカレントドライブ(ディレクトリ)に出力されます。
"X"を指定した場合はエラーメッセージを含むリストがコンソール上に出力されます。
"Z"を指定した場合はアセンブリリストファイルは作成されません。

s、h、pは3個を同時に指定してください。1個のみの指定は行えません。

−Hオプション: "−H"はメロディドキュメントファイルからメロディHEXファイルへの変換プログラムを起動させるオプションです。
このオプションを指定すると[shp]のオプションは無効となり、メロディドキュメントファイルの入力とメロディHEXファイルの作成はカレントドライブに対して行われます。
大文字/小文字の入力が可能です。

例: A>MLA628X C28XYYY↵

この例ではソースファイルC28XYYY.MDTがAドライブから入力され、メロディHEXファイルC28XYYYA.HEXとメロディアセンブリリストファイルC28XYYY.MPRおよびメロディドキュメントファイルC28XYYYA.DOCがAドライブに作成されます。

A>MLA628X B:C28XYYY↵

この例ではソースファイルC28XYYY.MDTがBドライブから入力され、メロディHEXファイルC28XYYYA.HEXとメロディアセンブリリストファイルC28XYYY.MPRおよびメロディドキュメントファイルC28XYYYA.DOCがBドライブに作成されます。

A>MLA628X C28XYYY.BBZ↵

この例ではソースファイルC28XYYY.MDTがBドライブから入力され、メロディHEXファイルC28XYYYA.HEXおよびメロディドキュメントファイルC28XYYYA.DOCがBドライブに作成されます。アセンブリリストファイルは作成されません。

MLA628Xが起動すると、以下の起動メッセージが表示されます。

```
A>MLA628X C28X0A0
          *** MLA628X MELODY ASSEMBLER. --- Ver 3.10 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN  NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP           SSS   SSS    OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS  SSS     OOO    OOO     NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO      NNN      NN

             (C) COPYRIGHT 1991 SEIKO EPSON CORP.

          SOURCE FILE NEME IS " C28XYYY.MDT "

          THIS SOFTWARE MAKES NEXT FILES.

             C28XYYYA.HEX  ...  MELODY HEX FILE.
             C28XYYYA.DOC  ...  MELODY DOCUMENT FILE.
             C28XYYY .MPR  ...  MELODY ASSEMBLY FILE.


                          STRIKE ANY KEY
```

例: C28X0A0.MDTのアセンブルを行う場合(オプション指定なし)

"STRIKE ANY KEY"の表示でプログラムは確認のキー入力を要求します。

任意のキー入力によりプログラムは先に進みます。
中止する場合はCTRL + Cを入力してください。DOSのコマンドレベルに戻ります。

```
A>MLA628X C28X0A0 -H
          *** MLA628X MELODY ASSEMBLER. --- Ver 3.10 ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN  NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP           SSS   SSS    OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS  SSS     OOO    OOO     NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO      NNN      NN

             (C) COPYRIGHT 1991 SEIKO EPSON CORP.

          SOURCE FILE NEME IS " C28XYYYA.DOC "

          THIS SOFTWARE MAKES NEXT FILES.

             C28XYYYA.HEX  ...  MELODY HEX FILE.


                          STRIKE ANY KEY
```

例: −Hオプション付きでC28X0A0.MDTのアセンブルを行う場合(メロディドキュメントファイルからメロディHEXファイルへの変換プログラムの起動)

"STRIKE ANY KEY"の表示でプログラムは確認のキー入力を要求します。入力したソースファイル名、オプションをもう一度確認してください。

任意のキー入力によりプログラムは先に進みます。
中止する場合はCTRL + Cを入力してください。DOSのコマンドレベルに戻ります。

4 ソースファイルの形式

ソースファイルの内容は以下に説明するメロディ記述記号と擬似命令で構成され、EDLINなどのエディタで作成します。

4.1 ソースファイル名

ソースファイルには7文字以下の任意の名前を付けることができますが、基本的には次の書式にしたがって付けてください。

C28XYYY.MDT

"YYY"は3文字以内の英数字で、セイコーエプソンから連絡する機種名を参考にしてください。
また、拡張子は必ず".MDT"にします。

4.2 ステートメント (行)

ソースファイルの各ステートメント(行)は次の書式で記述します。

基本ﾌｫｰﾏｯﾄ: **<アタック>** **<音符>** **<音階>** **<エンドビット>** **<コメント>**

例:

```
.TEMPC0=5
.TEMPC1=8
.OCTAVE=32
;
0           1           C3
:           :           :
0           6           A4#           1          ; 1st Melody
:           :           :
```

アタックフィールド　ノートフィールド　スケールフィールド　エンドビットフィールド　コメントフィールド

ステートメントはアタックフィールド、ノートフィールド、スケールフィールド、エンドビットフィールド、コメントフィールドの5つのフィールドで構成され、最大80文字まで記述できます。各フィールド間には1個以上のスペース、またはタブを挿入して区切りとします。
なお、エンドビットフィールド、コメントフィールドについては必要に応じて記述します。
CR(キャリッジリターン)コードのみの空白行も許されます。ただし、最終行に空白は許されません。各フィールドとも任意のカラムから開始できます。

(1) アタックフィールド

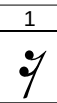
アタックフィールドにはアタック出力の制御を記述します。
"1"が記述された場合はアタック出力を行い、"0"が記述された場合はアタック出力を行いません。

(2) ノートフィールド

表4.2.1　音符と番号の対応

数字	1	2	3	4	5	6	7	8
音符								

表4.2.2　休符と番号の対応

数字	1	2	3	4	5	6	7	8
休符								

ノートフィールドには演奏させる音符を記述します。
1－8の数字によってメロディROMデータの音符データに対応する8種類の音符が指定できます。

また、スケールフィールドに"RR"(休符)を記述した場合は、この数字により休符を指定することができます。

(3) スケールフィールド

スケールフィールドには演奏させる音階(C3－C6#)を記述します。
スケールデータを直接入力する場合はデータの前に"$"を記述します。
直接入力する場合、データの範囲は00H～FDHです。
また、"RR"と記述することにより休符を選択することができます。
指定可能な音階の数は機種により異なります。("S5U1C62N8xD Manual"参照)

(4) エンドビットフィールド

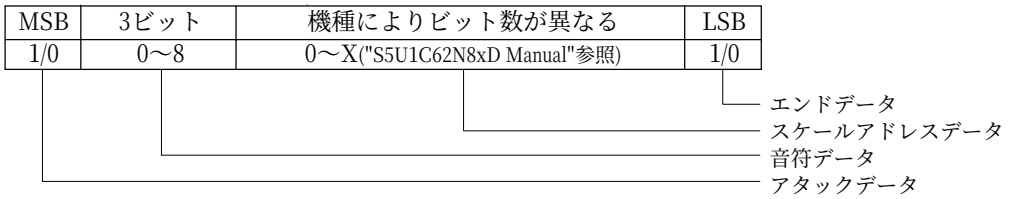
エンドビットフィールドには1曲の終了を示す命令を記述します。
"1"を記述することにより、そのアドレスのメロディデータで曲を終了します。
曲の終了以外は"0"を記述します。また、"0"記述の省略が可能です。

(5) コメントフィールド

コメントフィールドはプログラムの見出しや処理内容など、任意のコメントを記述するもので、アセンブルにより作成されるメロディHEXファイルには影響を与えません。
セミコロン";"から行末のCRコードまでがコメントフィールドとして見なされます。
コメントフィールドのみの行も許されますが、コメントが複数行にわたる場合でも各行のコメントの先頭にはセミコロンが必要です。

(6) 各フィールドとメロディデータの対応

* メロディデータ



- **エンドデータ**
  エンドビットフィールドが無記入または"0"の場合に"0"、"1"の場合は"1"となります。

- **スケールアドレスデータ**

表4.2.3　音階とスケールデータの対応

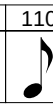
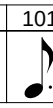
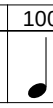
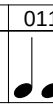
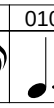
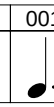
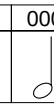
音階	スケールデータ								
	S7	S6	S5	S4	S3	S2	S1	S0	Hex.
C3	0	0	0	0	0	1	0	0	04
C3#	0	0	0	1	0	0	1	0	12
D3	0	0	1	0	0	0	0	0	20
D3#	0	0	1	0	1	1	1	1	2F
E3	0	0	1	1	1	0	1	1	3B
F3	0	1	0	0	0	1	0	0	44
F3#	0	1	0	1	0	0	0	1	51
G3	0	1	0	1	1	0	1	1	5B
G3#	0	1	1	0	0	1	0	1	65
A3	0	1	1	0	1	1	0	0	6C
A3#	0	1	1	1	0	1	0	0	74
B3	0	1	1	1	1	1	0	0	7C
C4	1	0	0	0	0	1	0	0	84
C4#	1	0	0	0	1	1	0	1	8D
D4	1	0	0	1	0	0	1	0	92
D4#	1	0	0	1	1	0	0	0	98
E4	1	0	0	1	1	1	1	0	9E
F4	1	0	1	0	0	1	0	0	A4
F4#	1	0	1	0	1	0	1	1	AB

音階	スケールデータ								
	S7	S6	S5	S4	S3	S2	S1	S0	Hex.
G4	1	0	1	1	0	0	0	1	B1
G4#	1	0	1	1	0	1	0	1	B5
A4	1	0	1	1	1	0	0	0	B8
A4#	1	0	1	1	1	1	0	0	BC
B4	1	1	0	0	0	0	0	0	C0
C5	1	1	0	0	0	1	0	0	C4
C5#	1	1	0	0	1	0	0	0	C8
D5	1	1	0	0	1	1	0	1	CD
D5#	1	1	0	0	1	1	1	0	CE
E5	1	1	0	1	0	0	1	1	D3
F5	1	1	0	1	0	1	0	0	D4
F5#	1	1	0	1	1	0	0	1	D9
G5	1	1	0	1	1	0	1	1	DB
G5#	1	1	0	1	1	1	0	0	DC
A5	1	1	0	1	1	1	1	0	DE
A5#	1	1	1	0	0	0	0	0	E0
B5	1	1	1	0	0	0	1	0	E2
C6	1	1	1	0	0	1	0	0	E4
C6#	1	1	1	0	0	1	1	0	E6

スケールフィールドに記述された音階またはスケールデータがスケールROMに格納され、そのアドレスがスケールアドレスデータとなります。

- **音符データ**

表4.2.4　音符とノートコードの対応

音符データ	111	110	101	100	011	010	001	000
音符								

音符と音符データの対応は表4.2.4のとおりです。

- **アタックデータ**
  アタックフィールドに記述された"0"、"1"がそのままアタックデータとなります。

5 疑似命令

疑似命令はアセンブラに対するもので、開発後のメロディ演奏時に実行されるものではありません。
以下の説明の中で疑似命令のフォーマットに使用されている< >は、ステートメント中に記述する内容を表します(< >自体は記述しません)。_は1個以上のスペースまたはタブを表します。また、<式>にはシンボル、定数、演算式などを記述します。

■ アドレス設定命令

- ORG (ORIGIN)

ORG_<式>	ロケーションカウンタの設定

ORG命令は<式>の値をロケーションカウンタに設定します。
ソースファイルの先頭にORG命令がない場合、ロケーションカウンタは0番地に設定され、アセンブルが行われます。
ORG命令はソースファイル内の複数箇所で使用することができます。ただし、現在のロケーションカウンタより前のロケーションに設定することはできません。このような指定を行った場合には、次に正しい設定が行われるまでのステートメントがすべて無効となり、"!"(Warning)が表示されます。
ROM容量を越えた値を指定した場合はRエラーになります。

■ オプション設定命令

テンポ選択

テンポは、16種類用意されており、この中から任意の2テンポを選択することができます。
選択の方法は、オプション設定命令(.TEMPC0=nと.TEMPC1=n)を使用します。
テンポ記号(♩≒k)とオプション設定命令との対応は表5.1のとおりで、nを指定することにより任意の2テンポが選択できます。
選択された2テンポのうちどちらを使用するかはプログラムによって決められ、プログラムでTEMPCレジスタ(アドレス: F2H、ビットアドレス:D1)に"0"を設定すると.TEMPC0=nで指定されたテンポが、"1"に設定すると.TEMPC1=nで指定されたテンポが使用されます。
テンポ選択はテンポが途中で変化する曲などに使用します。

表5.1 テンポの設定

テンポ記号	オプション設定命令	テンポ記号	オプション設定命令
♩≒ 30	.TEMPC0 = 0 .TEMPC1 = 0	♩≒ 60	.TEMPC0 = 8 .TEMPC1 = 8
♩≒ 32	.TEMPC0 = 1 .TEMPC1 = 1	♩≒ 68.6	.TEMPC0 = 9 .TEMPC1 = 9
♩≒ 34.3	.TEMPC0 = 2 .TEMPC1 = 2	♩≒ 80	.TEMPC0 = 10 .TEMPC1 = 10
♩≒ 36.9	.TEMPC0 = 3 .TEMPC1 = 3	♩≒ 96	.TEMPC0 = 11 .TEMPC1 = 11
♩≒ 40	.TEMPC0 = 4 .TEMPC1 = 4	♩≒ 120	.TEMPC0 = 12 .TEMPC1 = 12
♩≒ 43.6	.TEMPC0 = 5 .TEMPC1 = 5	♩≒ 160	.TEMPC0 = 13 .TEMPC1 = 13
♩≒ 48	.TEMPC0 = 6 .TEMPC1 = 6	♩≒ 240	.TEMPC0 = 14 .TEMPC1 = 14
♩≒ 53.3	.TEMPC0 = 7 .TEMPC1 = 7	♩≒ 480	.TEMPC0 = 15 .TEMPC1 = 15

- .TEMPC0

.TEMPC0=n	TEMPC0の設定 (n=0－15)

TEMPC0のオプション設定を行います。TEMP0－15のテンポの中から1つをnによって指定できます。この命令を省略することはできません。

- .TEMPC1

.TEMPC1=n	TEMPC1の設定 (n=0－15)

TEMPC1のオプション設定を行います。TEMP0－15のテンポの中から1つをnによって指定できます。この命令を省略することはできません。

• .OCTAVE

.OCTAVE=m　　音域の設定 (m=32または64)

メロディ逓倍回路の仕様を指定し、音域を決定します。
m=32の場合は32kHz仕様となり、音域はC3－C6#となります。
m=64の場合は64kHz仕様となり、音域は32kHz仕様に比べて1オクターブ高いC4－C7#となります。
この場合、ソースファイルの表現上C5の音階であっても実際にはC6の音を発生します。
この命令を省略することはできません。

6 エラーメッセージ

アセンブル中にエラーが発生した場合、MLA628Xは以下のエラーシンボル、あるいはエラーメッセージをコンソールとアセンブリリストファイルに出力します。

エラーシンボルはエラーが発生したステートメントの先頭(第1カラム)に1つだけ(複数のエラーがある場合、優先順位の高いものが)出力されます。

■ エラーシンボル (アセンブル可能なエラー)

以下のエラーシンボルは優先順位の高いものから順に示しています。

- S (Syntax Error) ..... 文法上で重大な誤りがある
  スケールフィールドにおけるエラー　　: 音階の範囲(C3－C6#)を外れた指定
  ノートフィールドにおけるエラー　　: 音符の範囲(1－8)を外れた指定
  アタックフィールドにおけるエラー　　: 0/1以外の数を指定
  エンドビットフィールドにおけるエラー: 0/1以外の数を指定
- O (Scale ROM Over Flow) ..... スケールROM容量を越えて定義した
- R (Range Error) ..... ロケーションカウンタの値がメロディROM容量の上限を越えたまたは上限を越えるロケーションを指定した

■ エラーメッセージ (アセンブルしない、またはアセンブル結果を出力しないフェータルエラー)

- OPTION COMMAND MISSING ..... オプション設定が行われていない
- FILE NAME ERROR ソースファイル名が8文字以上である
- FILE NOT PRESENT ..... 指定されたソースファイルが存在しない
- DIRECTORY FULL ..... 指定ディスクのディレクトリに余裕がない
- FATAL DISK WRITE ERROR ..... ディスク上にファイルが書き込めない

Appendix 入出力ファイル例

以下の入出力ファイルはMLA6282の場合の例で、機種によりデータのサイズ等が異なります。

■ ソースファイル例

```
.TEMPC0=5
.TEMPC1=8
.OCTAVE=32
;
1 1 C3     ; 𝅘𝅥𝅯
0 4 D4     ; 𝅘𝅥
0 4 F4     ; 𝅘𝅥
0 2 F5     ; 𝅘𝅥𝅮
0 3 G5#    ; 𝅘𝅥𝅮.
1 7 A4     ; 𝅘𝅥..
1 5 B4     ; 𝅘𝅥𝅯𝅯
0 6 A4# 1  ; 𝅘𝅥.   1st Melody
;
ORG 10H
;
1 2 $C3    ; 𝅘𝅥𝅮
0 3 $45    ; 𝅘𝅥𝅮.
0 7 $E3    ; 𝅘𝅥..
1 6 $97    ; 𝅘𝅥.
0 5 C6     ; 𝅘𝅥𝅯𝅯
0 7 A5#    ; 𝅘𝅥..
1 3 $42 1  ; 𝅘𝅥𝅮   2nd Melody
```

■ アセンブリリスト例

```
LISTING OF MLA6282      C282YYY.MPR  1991-6-01 14:25...PAGE   1

  ADRS  CODE      SOURCE

                .TEMPC0 = 5
                .TEMPC1 = 8
                .OCTAVE = 32
                ;
  00    3C0      1 1 C3     ; 𝅘𝅥𝅯
  01    102      0 4 D4     ; 𝅘𝅥
  02    104      0 4 F4     ; 𝅘𝅥
  03    186      0 2 F5     ; 𝅘𝅥𝅮
  04    148      0 3 G5#    ; 𝅘𝅥𝅮.
  05    24A      1 7 A4     ; 𝅘𝅥..
  06    2CC      1 5 B4     ; 𝅘𝅥𝅯𝅯
  07    08F      0 6 A4# 1  ; 𝅘𝅥.   1st Melody
                ;
                ORG 10H
                ;
  10    390      1 2 $C3    ; 𝅘𝅥𝅮
  11    152      0 3 $45    ; 𝅘𝅥𝅮.
  12    054      0 7 $E3    ; 𝅘𝅥..
  13    296      1 6 $97    ; 𝅘𝅥.
  14    0D8      0 5 C6     ; 𝅘𝅥𝅯𝅯
  15    05A      0 7 A5#    ; 𝅘𝅥..
  16    35D      1 3 $42 1  ; 𝅘𝅥𝅮   2nd Melody

0 ERROR(S) DETECTED
```

時間
日付け
メロディアセンブリリスト
ファイル名

```
SCALE ROM TABLE      PAGE S-1

ADRS     SCALE       CODE

00000      C3          04
00001      D4          92
00010      F4          A4
00011      F5          D4
00100      G5#         DC
00101      A4          B8
00110      B4          C0
00111      A4#         BC
01000      $C3         C3
01001      $45         45
01010      $E3         E3
01011      $97         97
01100      C6          E4
01101      A5#         E0
01110      $42         42
01111      --          FF
10000      --          FF
10001      --          FF
10010      --          FF
10011      --          FF
10100      --          FF
10101      --          FF
10110      --          FF
10111      --          FF
11000      --          FF
11001      --          FF
11010      --          FF
11011      --          FF
11100      --          FF
11101      --          FF
11110      --          FF
11111      RR          C4
```

スケールROMテーブルの例
- 未使用のコードは"--"(ハイフン)が表示されます。
- 未使用の場合、CODEはFFHとなります。
- スケールROMの最終ADRS="11111"(S1C62N82の場合)は、SCALE="RR"、CODE="C4"に固定されます。

■ メロディHEXファイルデータフォーマット例

```
:10000000010101010102020 0FFFFFFFFFFFFFFFEF
:10001000010100020000003FFFFFFFFFFFFFFFFE2
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:10008000C0020486484ACC8FFFFFFFFFFFFFFFF3F
:1000900090525496D85A5DFFFFFFFFFFFFFFFF0E
:1000A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF60
:1000B000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF50
:1000C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF40
:1000D000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF30
:1000E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF20
:1000F000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF10
:100100000492A4D4DCB8C0BCC345E397E4E042FF4A
:10011000FFFFFFFFFFFFFFFFFFFFFFFFFFFFC42A
:100120000508FFFFFFFFFFFFFFFFFFFFFFFFFFD0
:1001300000FFFFFFFFFFFFFFFFFFFFFFFFFFFFCE
:00000001FF
```

メインROM上位 (D8, D9)

メインROM下位 (D0－D7)

スケールROM (D0－D7)

オプション (テンポ、逓倍)

■ エラーが発生した場合のアセンブリリスト例

エラーが発生した行のコードは強制的にFFFHになり、スケールROMデータも作成されません。

```
LISTING OF MLA6282      C282YYY.MPR  1991-6-01 17:30...PAGE   1

  ADRS  CODE      SOURCE

                .TEMPC0 = 5
                .TEMPC1 = 8
                .OCTAVE = 32
                ;
   0    000      0 1 C3    ; 𝅘𝅥𝅯
   1    062      0 4 D4    ; ♩
   2    064      0 4 F4    ; ♩
S  3    FFF      0 2 F6    ; ♪
   4    048      0 3 G5#   ; ♪.
S  5    FFF      2 7 A4    ; ♩..
S  6    FFF      1 9 B4
   7    0A9      0 6 A4# 1 ; ♩.  1st Melody
                ;
                ORG 10H
                ;
  10    02A      0 2 $C3   ; ♪
  11    06C      0 3 $17   ; ♪.
  12    0CE      0 7 $E3   ; ♩..
  13    1B0      1 6 $97   ; ♩.
  14    092      0 5 C6    ; ♩𝅘𝅥𝅯
  15    0D4      0 7 A5#   ; ♩..
  16    056      0 3 E3    ; ♪.
  17    078      0 4 G5    ; ♩
  18    09A      0 5 G4    ; ♩𝅘𝅥𝅯
  19    07C      0 4 G4#   ; ♩
O 1A    FFF      0 3 A4    ; ♪.
  1B    15F      1 3 RR 1  ; ♪  2nd Melody
                  .
                  .
                  .
  4E    05A      0 3 G4    ; ♪.
  4F    05A      0 3 G4    ; ♪.
R 50    FFF      0 3 G4    ; ♪.
S 51    FFF      4 3 G4    ; ♪.
R 52    FFF      0 3 G4 1  ; ♪.

7 ERROR(S) DETECTED
```

```
SCALE ROM TABLE      PAGE S-1

ADRS      SCALE      CODE

00000      C3          04
00001      D4          92
00010      F4          A4
00011      F5          D4
00100      G5#         DC
00101      A4          B8
00110      B4          C0
00111      A4#         BC
01000      $C3         C3
01001      $45         45
01010      $E3         E3
01011      $97         97
01100      C6          E4
01101      A5#         E0
01110      $42         42
01111      --          FF
10000      --          FF
10001      --          FF
10010      --          FF
10011      --          FF
10100      --          FF
10101      --          FF
10110      --          FF
10111      --          FF
11000      --          FF
11001      --          FF
11010      --          FF
11011      --          FF
11100      --          FF
11101      --          FF
11110      --          FF
11111      RR          C4
```

■ ドキュメントファイルフォーマット例

```
:1000000001010101020200FFFFFFFFFFFFFFEF
:1000100001010002000003FFFFFFFFFFFFFFFFE2
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:00000001FF
```

メインROM (HIGH側)
インテルヘキサフォーマット

```
:10000000C0020486484ACC8FFFFFFFFFFFFFFFBF
:1000100090525496D85A5DFFFFFFFFFFFFFFFF8E
:10002000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE0
:10003000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD0
:10004000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFC0
:10005000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFB0
:10006000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFA0
:10007000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF90
:00000001FF
```

メインROM (LOW側)
インテルヘキサフォーマット

```
:100000000492A4D4DCB8C0BCC345E397E4E042FF4B
:10001000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFC42B
:00000001FF
```

スケールROM
インテルヘキサフォーマット

```
*   E0C6282 MELODY OPTION DOCUMENT V.3.10
*
* OPTION NO.20
*   OCTAVE CIRCUIT
*    32KHZ -------------------- SELECTED
OPT2001 01
*
* OPTION NO.21
*   < MELODY TEMPO SELECTION >
*   TEMPC0 TEMPO 5 ------------ SELECTED
*   TEMPC1 TEMPO 8 ------------ SELECTED
OPT2101 03
OPT2102 04
OPT2103 02
OPT2104 04
¥¥END
```

オプション選択

V

Function Option Generator

FOG62XX

ここでは4-bit Single Chip Microcomputer S1C62XXXの
ファンクションオプションを設定するソフトウェア
ツールFOG62XXの操作方法について解説しています。

Function Option Generator FOG62XX

- 目 次 -

1 機種による相違点 V-1

2 FOG62XXの概要 V-1

2.1 概要と実行フロー V-1

2.2 FOG62XXの入出力ファイル V-1

3 オプションリストの作成 V-2

3.1 オプションリストの記入方法 V-2

3.2 オプションリスト例 V-2

4 FOG62XXの操作方法 V-3

4.1 FOG62XXの起動 V-3

4.2 ファンクションオプションの新規設定 V-4

4.3 ファンクションオプションの設定変更 V-5

4.4 オプション仕様の選択 V-6

4.5 HEXファイルの作成とEPROMの選択 V-6

4.6 終了方法 V-7

1 機種による相違点

各機種により設定されているオプションの内容が異なります。
ここでは操作方法の解説のみを行いますので、オプション仕様およびその選択画面については"S5U1C62xxxD Manual"を参照してください。

2 FOG62XXの概要

2.1 概要と実行フロー

4-bit Single Chip Microcomputer S1C62XXXはI/Oポートの機能などをハードウェアオプションとして選択できるようになっており、S1C62XXXのマスクパターンをオプション選択にしたがって変更することにより、ターゲットシステムの仕様に合わせた機能に設定できます。
Function Option Generator FOG62XX(以後は単にFOG62XXと記述します)は、このマスクパターン生成のためのデータファイルを作成するソフトウェアツールで、ハードウェアオプション1つひとつの設定内容が対話形式で選択できます。FOG62XXで作成されたデータファイルから、S1C62XXXのマスクパターンが大型コンピュータにより自動生成されます。
また、Evaluation Board(S5U1C62xxxE)のハードウェアオプションROM用のHEXファイルが同時に作成され、その内容をEPROMに書き込みEvaluation Boardに装着することによって、Evaluation Board上でオプション機能が実現できます。

ファンクションオプションジェネレータのプログラム名は次のとおりです。

FOG62XX. EXE

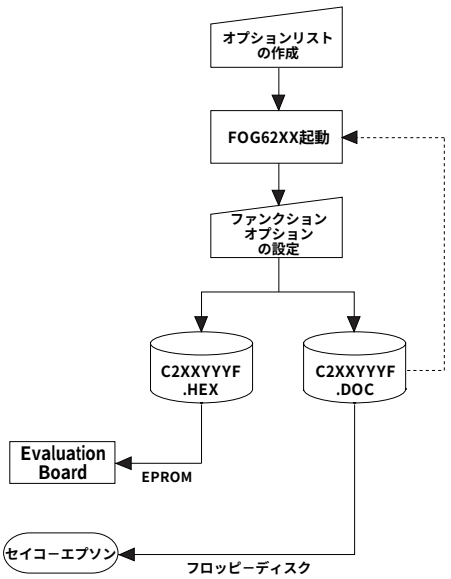


図2.1.1 FOG62XXの実行フロー

図2.1.1にFOG62XXの実行フローを示します。

2.2 FOG62XXの入出力ファイル

オプションの設定はプログラム内で対話形式で行えるため、ソースファイルは必要ありません。その代わりに、ターゲットシステムの仕様に合ったハードウェアオプションをFOG62XX実行前に選択し、オプションリスト(入力の準備として記入する用紙、後述)に記入しておきます。
FOG62XXの実行後は以下のデータファイルが生成されます。

■ **ファンクションオプションドキュメントファイル (C2XXYYYF.DOC)**
I/Oポートなどのマスクパターンを作成するためのデータファイルです。このファイルをFOG62XXで入力して内容の一部を修正することもできます。

■ **ファンクションオプションHEXファイル (C2XXYYYF.HEX)**
Evaluation Board用のファンクションオプションファイル(インテルHEX形式)です。このファイルをROMライタによって書き込み、Evaluation Boardのファンクションオプション ROMを1個作成します。

＊ ファイル名の"YYY"はお客さまごとにセイコーエプソンが指定させていただきます。

＊ ファンクションオプションドキュメントファイルはマスクデータチェッカ(MDC62XX)を用いてプログラムファイル(C2XXYYYH.HEX、C2XXYYYL.HEX)およびセグメントオプションドキュメントファイル(C2XXYYYS.DOC)と一括して他のディスケットにコピーし、セイコーエプソンに提出してください。

＊ HEXファイルをEPROMに書き込む際は、ROMの未使用領域をすべてFFHに設定してください。ROMの登載位置については"S5U1C62xxxE Manual"を参照してください。

3 オプションリストの作成

3.1 オプションリストの記入方法

各機種のオプションリストおよびオプション仕様の解説は"S5U1C62xxxD Manual"に掲載されています。次項のオプションリスト例に示されるとおり、オプション項目には複数の仕様が用意されています。"S5U1C62xxxD Manual"のオプション仕様解説を参照のうえターゲットシステムに合った仕様を選択し、□に✔印を記入してください。使用しないポートなどについても、解説にしたがって必ず記入してください。このオプションリストを参照して、オプション選択を画面上で行います。

3.2 オプションリスト例

以下にオプションリストの例を示します。各機種のオプションリストについては"S5U1C62xxxD Manual"を参照してください。

1 機種 . . . DEVICE TYPE
□ 1. E0C62XX
□ 2. E0C62LXX

2 キー同時押しリセット . . . MULTIPLE KEY ENTRY RESET
・組み合わせ ..... □ 1. 使用しない
□ 2. 使用する K00, K01
□ 3. 使用する K00, K01, K02
□ 4. 使用する K00, K01, K02, K03

3 入力割り込み ノイズリジェクト . . . INTERRUPT NOISE REJECTOR
・K00～K03 ..... □ 1. 使用する □ 2. 使用しない

4 入力ポート プルダウン抵抗 . . . INPUT PORT PULL DOWN RESISTOR
・K00 ..... □ 1. あり □ 2. なし
・K01 ..... □ 1. あり □ 2. なし
・K02 ..... □ 1. あり □ 2. なし
・K03 ..... □ 1. あり □ 2. なし

5 R00 端子仕様 . . . R00 SPECIFICATION
・出力形式 ..... □ 1. DC出力
□ 2.ブザー反転出力(R00ポートによる制御)
□ 3.ブザー反転出力(R01ポートによる制御)
・出力仕様 ..... □ 1. コンプリメンタリ □ 2. Pchオープンドレイン

6 R01 端子仕様 . . . R01 SPECIFICATION
・出力形式 ..... □ 1. DC出力 □ 2. ブザー出力
・出力仕様 ..... □ 1. コンプリメンタリ □ 2. Pchオープンドレイン

7 出力ポート(R02, R03)出力仕様 . . . OUTPUT PORT SPECIFICATION (R02, R03)
・R02 ..... □ 1. コンプリメンタリ □ 2. Pchオープンドレイン
・R03 ..... □ 1. コンプリメンタリ □ 2. Pchオープンドレイン

:

4 FOG62XXの操作方法

4.1 FOG62XXの起動

FOG62XXを起動させるにはDOSのコマンドレベル(A>などのプロンプトが表示されている状態)で次のプログラム名を入力します。DMS6200から起動させる場合は、メニュー画面で"FOG62XX.EXE"を選択してください。

```
A>FOG62XX↵
```

↵はリターンキーの入力を表します。

FOG62XXが起動すると、次の起動メッセージが表示されます。

```
     ***  E0C62XX FUNCTION OPTION GENERATOR. --- Ver 3.02  ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO      NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS      SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP           SSS    SSS    OOO      OOO   NNN    NNNN
EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO      NNN      NN

             (C) COPYRIGHT 1991  SEIKO EPSON CORP.

         THIS SOFTWARE MAKES NEXT FILES.

            C2XXYYYF.HEX  ...  FUNCTION OPTION HEX FILE.
            C2XXYYYF.DOC  ...  FUNCTION OPTION DOCUMENT FILE.

                       STRIKE ANY KEY.
```

"STRIKE ANY KEY"が表示されたところで任意のキー入力を行い、プログラムを先に進めます。
中止する場合はCTRL+C(使用するPCによりSTOPも可)を入力してください。DOSのコマンドレベルに戻ります。

起動メッセージの後、画面には現在パソコンに設定されている日付が表示され、新たな日付の入力を要求します。

```
*** E0C62XX USER'S OPTION SETTING. --- Ver 3.02 ***

CURRENT DATE IS  91/07/19
PLEASE INPUT NEW DATE  :   91/07/22↵
```

日付を変更する場合は年月日をそれぞれ2桁ずつ、"/"で区切って入力します。
日付を変更しない場合は"↵"キーのみを入力して、先に進めてください。

日付けの設定が終わると、次の操作選択メニューが表示されます。

```
*** OPERATION SELECT MENU ***

       1. INPUT NEW FILE
       2. EDIT FILE
       3. RETURN TO DOS

PLEASE SELECT NO.?
```

ここで1～3の番号を入力して以後の操作を選択します。

それぞれの内容は以下のとおりです。

1. **INPUT NEW FILE:** ファンクションオプションの新規設定を行います。
2. **EDIT FILE:** すでに作成してあるファンクションオプションドキュメントファイルを読み込み、オプションの設定変更を行います。
   この場合、作業ディスク上に"1. INPUT NEW FILE"によって作成されたファンクションオプションドキュメントファイル(C2XXYYYF.DOC)が必要です。
3. **RETURN TO DOS:** FOG62XXを終了して、DOSのコマンドレベルに戻ります。

4.2 ファンクションオプションの新規設定

以下にファンクションオプションを新規設定する場合の操作方法を示します。

＊ 例文中の↲はリターンキーの入力を表します。

```
*** OPERATION SELECT MENU ***

        1. INPUT NEW FILE
        2. EDIT FILE
        3. RETURN TO DOS

PLEASE SELECT NO.? 1↲                                                    .. (1)

PLEASE INPUT FILE NAME? C2XXYYY↲                                         .. (2)
PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↲                             .. (3)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↲                               .. (4)
                    ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↲
                    ? ↲
```

(1) PLEASE SELECT NO.?

操作選択メニューで、"1. INPUT NEW FILE"を選択します。

(2) PLEASE INPUT FILE NAME?

これから作成するファンクションオプションのドキュメント(HEX)ファイル名を入力します(エクステンションは入力しないでください)。
カレントドライブ上に指定した名前のファイルが存在する場合、次例のようなメッセージが表示されます。"Y"を入力すると旧ファイルに新データを上書きし、"N"を入力すると再度ファイル名の入力状態になります。

```
PLEASE INPUT FILE NAME? C2XXYYY↲
EXISTS OVERWRITE(Y/N)?
```

(3) PLEASE INPUT USER'S NAME?

お客さまの会社名を入力します。

(4) PLEASE INPUT ANY COMMENT

コメントを入力します。
1行に入力可能な文字数は50文字まで(それ以上の入力は50文字以降が削除されます)で、最大10行まで入力することができます。コメントの入力を終了する場合は必要な行数の入力を行い、次の行頭でリターンキーを入力します。

コメントには次のような内容を含めるようお願いします。

- 事業所、所属
- 所在地、電話番号、FAX番号
- その他、技術情報等

以上の入力が終了するとファンクションオプションの設定作業に入ります。
新規作成の場合、No.1から最後までのファンクションオプションを順次、対話形式で選択していきます。
ファンクションオプションそれぞれの選択方法については"S5U1C62xxxD Manual"を参照してください。

4.3 ファンクションオプションの設定変更

以下にファンクションオプションの設定を変更する場合の操作方法を示します。

＊ 例文中の↵はリターンキーの入力を表します。

```
*** OPERATION SELECT MENU ***

        1. INPUT NEW FILE
        2. EDIT FILE
        3. RETURN TO DOS

PLEASE SELECT NO.? 2↵                                                  .. (1)

*** SOURCE FILE(S) ***

C2XX0A0         C2XX0B0         C2XX0C0                                .. (2)

PLEASE INPUT FILE NAME? C2XXYYY↵                                       .. (3)
PLEASE INPUT USER'S NAME?↵                                             .. (4)
PLEASE INPUT ANY COMMENT
(ONE LINE IS 50 CHR)?↵                                                 .. (5)
PLEASE INPUT EDIT NO.? 4↵                                              .. (6)
```

(1) PLEASE SELECT NO.?

操作選択メニューで"2. EDIT FILE"を選択します。

(2) * SOURCE FILE(S) *****

カレントドライブ上で変更可能なファイルの一覧を表示します。
変更可能なファイルが存在しない場合は次のメッセージを表示してプログラムを終了します。

```
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

(3) PLEASE INPUT FILE NAME?

ファイル名を入力します(エクステンションは入力しないでください)。
カレントドライブ上に指定した名前のファンクションオプションドキュメントファイル(C2XXYYYF.DOC)が存在しない場合、次例のようにエラーメッセージが表示され、再度ファイル名の入力状態になります。

```
PLEASE INPUT FILE NAME? C2XX0N0↵
FUNCTION OPTION DOCUMENT FILE IS NOT FOUND.
```

(4) PLEASE INPUT USER'S NAME?

お客さまの会社名を変更する必要がある場合、再度入力し直すことによって変更が可能です。
通常はリターンキーのみの入力により、前回入力した内容をそのまま用います。

(5) PLEASE INPUT ANY COMMENT

コメントを変更する場合は最初の行より入力し直します(途中の行のみの変更は行えません)。
リターンキーのみの入力により、前回入力した内容をそのまま用いることができます。
入力条件は、新規設定時と同様です。

(6) PLEASE INPUT EDIT NO.?

設定を変更するファンクションオプションのNo.を入力します。No.が入力されるとオプションの設定作業に入ります。1つのオプションを選択し終わると再びNo.の入力状態になりますので、変更するすべてのオプションの選択を終了するまで同じ操作を繰り返します。このときにNo.を入力せずにリターンキーのみの入力を行うと次のNo.のオプションが選択できます。

オプションの設定を終了する場合は "E↵"を入力します。HEXファイル作成の確認操作(4.5項を参照)に移ります。

例: ・No.9のファンクションオプションの設定を変更する場合

```
PLEASE INPUT EDIT NO.? 9↵
```

・設定を終了させる場合

```
PLEASE INPUT EDIT NO.? E↵
```

4.4 オプション仕様の選択

各オプションの選択は対話形式で行います。新規設定の場合はNo.1から最後までのオプションすべてを順次設定し、設定変更の場合は指定したNo.のオプションを直接設定できます。

```
*** OPTION NO.3 ***

--- MULTIPLE KEY ENTRY RESET ---

        COMBINATION     1. NOT USE
                        2. USE  K00,K01
                        3. USE  K00,K01,K02
                        4. USE  K00,K01,K02,K03

PLEASE SELECT NO.(1)? 2↵

        2. USE  K00,K01   SELECTED
```

オプションの選択方法
各オプションの選択項目が番号で表示されますので、選択する内容を示す番号を入力します。

( )内の数字はデフォルト番号で、リターンキーのみを入力した場合にその項目が選択されます。

選択された項目は再度表示されます。

```
*** OPTION NO.4 ***

--- INTERRUPT NOISE REJECTOR ---

        K00-K03         1. USE
                        2. NOT USE

PLEASE SELECT NO.(1) ? B↵

*** OPTION NO.3 ***

--- MULTIPLE KEY ENTRY RESET ---

        COMBINATION     1. NOT USE
                        2. USE  K00,K01
                        3. USE  K00,K01,K02
                        4. USE  K00,K01,K02,K03

PLEASE SELECT NO.(1)?
```

新規設定中に選択を誤った場合は、次のオプション選択のところで"B↵"を入力すると再度選択し直すことができます。

新規設定ですべてのオプションの設定が終了するとEPROM選択に移ります。
設定変更でオプションの修正が済んだ場合は"E↵"を入力するとEPROM選択に移ります。

4.5 HEXファイルの作成とEPROMの選択

```
END OF OPTION SETTING.
 DO YOU MAKE HEX FILE (Y/N) ? Y↵                    ..... (1)

*** OPTION EPROM SELECT MENU ***

        1. 27C64
        2. 27C128
        3. 27C256
        4. 27C512

PLEASE SELECT NO.? 2↵                               ..... (2)

        2. 27C128   SELECTED
```

ファンクションオプションの設定が終了するとHEXファイル作成の有無について問い合わせてきます。
Evaluation Boardを使用してプログラムのデバッグを行う場合、HEXファイル(C2XXYYYF.HEX)が必要となります。

注: Evaluation Boardに装着するEPROMは以下の規格のものをご使用ください。

TACC ≦ 250nsec (TACC: アクセスタイム)

(1) DO YOU MAKE HEX FILE (Y/N) ?

HEXファイル(C2XXYYYF.HEX)を作成する場合は"Y"を入力します。"N"を入力した場合はHEXファイルは作成されず、ドキュメントファイル(C2XXYYYF.DOC)のみが作成されます。

(2) PLEASE SELECT NO.?

(1) で"Y"を入力した場合、オプションROM選択メニューが表示されますので、Evaluation Boardのオプション設定に使用するEPROMを選択します。(1) で"N"を入力した場合、このメニューは表示されません。
EPROMはファンクションオプション設定用に1個必要です。上記の例では"27C128"を選択しています。

```
MAKING FILE(S) IS COMPLETED.
```

ここまでの操作が終了するとFOG62XXは各ファイルを作成します。その後、左のメッセージが表示され、操作選択メニューに戻ります。

4.6 終了方法

以下に、FOG62XXを終了させる場合の操作方法を示します。

```
*** OPERATION SELECT MENU ***

        1. INPUT NEW FILE
        2. EDIT FILE
        3. RETURN TO DOS

PLEASE SELECT NO.? 3↵
```

一連の操作が終了すると操作選択メニューに戻ります。ここで、"3. RETURN TO DOS"を選択することによりFOG62XXを終了させることができます。
また、"1. INPUT NEW FILE"か"2. EDIT FILE"を選択した場合、再度ファンクションオプションの設定作業を行うことができます。

上記以外の方法として、プログラム実行中にCTRL+C(使用するPCによりSTOPも可)を入力することにより強制的に終了させることができます。

VI

Segment Option Generator

SOG62XX

ここでは4-bit Single Chip Microcomputer S1C62XXXの
セグメントオプションを設定するソフトウェアツール
SOG62XXの操作方法について解説しています。

Segment Option Generator SOG62XX

- 目　次 -

1 機種による相違点 VI-1

2 SOG62XXの概要 VI-1

2.1 概要と実行フロー VI-1

2.2 SOG62XXの入出力ファイル VI-1

3 オプションリスト VI-2

3.1 オプションリスト例 VI-2

3.2 セグメント出力ポート仕様 VI-2

4 SOG62XXの操作方法 VI-3

4.1 セグメントオプションソースファイルの作成 VI-3

4.2 SOG62XXの起動 VI-5

4.3 入力ファイルの選択 VI-5

4.4 HEXファイルの作成とEPROMの選択 VI-7

4.5 終了方法 VI-7

5 エラーメッセージ VI-8

1 機種による相違点

機種によりセグメント出力の仕様、表示メモリの容量とアドレスが異なります。
ここでは操作方法を中心に解説します。オプション仕様については4端子のコモン出力が設定されている標準的なセグメント仕様の機種の場合を例に概要説明を行いますので、各機種の詳細については"S5U1C62xxxD Manual"を参照してください。
なお、セグメントオプションが設定されていない機種のソフトウェアツールにはSOG62XXが含まれません。

2 SOG62XXの概要

2.1 概要と実行フロー

4-bit Single Chip Microcomputer S1C62XXXはLCDセグメントの構成をハードウェアオプションとして選択できるようになっており、S1C62XXXのマスクパターンをオプション選択にしたがって変更することにより、ターゲットシステムの仕様に合わせた機能に設定できます。
Segment Option Generator SOG62XX(以後は単にSOG62XXと記述します)は、このマスクパターン生成のためのデータファイルを作成するソフトウェアツールです。SOG62XXで作成されたデータファイルから、S1C62XXXのマスクパターンが大型コンピュータにより自動生成されます。
また、Evaluation Board(S5U1C62xxxE)のハードウェアオプションROM用のHEXファイルが同時に作成され、その内容をEPROMに書き込みEvaluation Boardに装着することによって、Evaluation Board上でオプション機能が実現できます。

セグメントオプションジェネレータのプログラム名は次のとおりです。

SOG62XX. EXE

図2.1.1にSOG62XXの実行フローを示します。

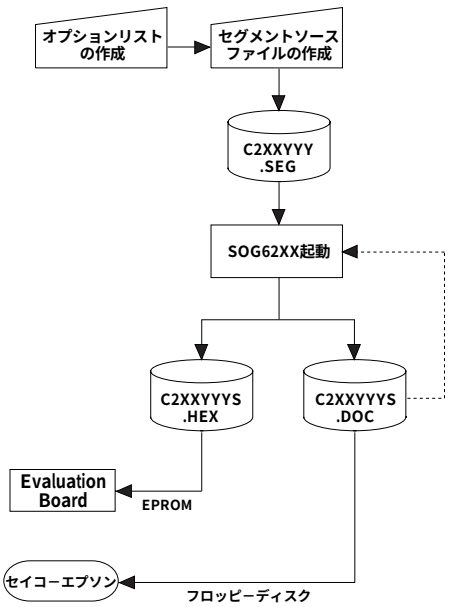


図2.1.1 SOG62XXの実行フロー

2.2 SOG62XXの入出力ファイル

SOG62XXはオプションの設定内容を記述したソースファイルを入力し、以下のデータファイルを出力します。

■ **セグメントオプションソースファイル (C2XXYYY.SEG)**
セグメントオプションの設定内容を記述します。エディタ等で作成します。

■ **セグメントオプションドキュメントファイル (C2XXYYYS.DOC)**
セグメントデコーダやセグメント出力ポートのマスクパターンを作成するためのデータファイルです。

■ **セグメントオプションHEXファイル (C2XXYYYS.HEX)**
Evaluation Board用のセグメントオプションファイル(インテルHEX形式)です。このファイルをROMライタによって書き込み、Evaluation Boardのセグメントオプション ROMを作成します。

＊ ファイル名の"YYY"はお客さまごとにセイコーエプソンが指定させていただきます。

＊ セグメントオプションドキュメントファイルはマスクデータチェッカ(MDC62XX)を用いてプログラムファイル(C2XXYYYH.HEX、C2XXYYYL.HEX)およびファンクションオプションドキュメントファイル(C2XXYYYF.DOC)と一括して他のディスケットにコピーし、セイコーエプソンに提出してください。

＊ HEXファイルをEPROMに書き込む際は、ROMの未使用領域をすべてFFHに設定してください。ROMの登載位置については"S5U1C62xxxE Manual"を参照してください。

3 オプションリスト

3.1 オプションリスト例

以下にオプションリストの例(4コモンの場合)を示します。各機種のオプションリストについては"S5U1C62xxxD Manual"を参照してください。

オプションリスト例

端子名	アドレス												出力仕様
	COM0			COM1			COM2			COM3			
	H	L	D	H	L	D	H	L	D	H	L	D	
SEG0													SEG出力 ☐
SEG1													DC出力 ☐ C ☐ P
SEG2													SEG出力 ☐
SEG3													DC出力 ☐ C ☐ P
SEG4													SEG出力 ☐
SEG5													DC出力 ☐ C ☐ P
:													: :

表中の記号は以下のとおりです。

〈アドレス〉 **H:** ページ内の上位アドレス
L: ページ内の下位アドレス (0〜F)
D: データビット (0〜3)

〈出力仕様〉 **C:** コンプリメンタリ出力
P: Pchオープンドレイン出力

例に示されるとおり、オプションリストにはセグメントに割り付ける表示メモリのアドレスおよびデータビットの記入欄と出力仕様の選択項目が用意されています。"S5U1C62xxxD Manual"のオプション仕様解説を参照のうえターゲットシステムに合った仕様を選択し、☐に✔印を記入してください。なお、表示メモリアドレスも記入してください。
このオプションリストを参照して、ソースファイルを作成します。

3.2 セグメント出力ポート仕様

セグメント出力ポート(SEG0〜SEG*)は2端子ごとにセグメント出力とDC出力の選択が可能です。オプションリストの出力仕様はこの選択項目を示しています。液晶パネルの駆動に用いる場合はセグメント出力、通常の出力ポートとして用いる場合にはDC出力を選択します。
DC出力を選択した場合は、さらにコンプリメンタリ出力とPchオープンドレイン出力(機種によってはNchオープンドレイン出力)のいずれかを選択できます。
なお、使用しないセグメント出力ポートはセグメント出力を選択してください。

■ セグメント出力を選択した場合

セグメント出力ポートはセグメントデコーダを内蔵しており、セグメントメモリ領域の任意のアドレス、データビットを任意のセグメントに割り付けることができます。この割り付け内容をオプションリストのアドレス欄に記入します。セグメントメモリ1ビットに対して割り付け可能なセグメントは1つのみで、複数の設定を行うことはできません。セグメント割り付けはHにページ内の上位アドレス、Lに下位アドレス(0〜F)、Dにデータビット(0〜3)を設定し、オプションリストのそれぞれの欄に記入します。使用しないセグメント端子はCOM0〜COM3のH、L、D欄に"---"(ハイフン)を記入してください。
このセグメントメモリのビットを1に設定すると割り付けられたセグメントが点灯し、0にすると消灯します。

■ DC出力を選択した場合

DC出力は2端子ごとに選択できます。また、コンプリメンタリ出力かオープンドレイン出力かの選択も同様に2端子ごとになります。
割り付けられたセグメントメモリのビットを1に設定するとセグメント出力ポートはHIGHレベルになり、0にするとLOWレベルになります。
セグメント割り付けはセグメント出力を選択した場合と同様ですが、出力制御はCOM0に割り付けられたセグメントメモリが有効となり、COM1〜COM3に割り付けられたセグメントメモリは無効となります。オプションリストのCOM1〜COM3には"---"(ハイフン)を記入してください。

注: コモン端子(COM0〜COM3)は機種により構成が異なることがあります。

4 SOG62XXの操作方法

4.1 セグメントオプションソースファイルの作成

SOG62XXは入力ファイルとしてセグメント出力ポートの仕様を記述したセグメントオプションソースファイルを必要とします。オプションリストに必要事項を記入後、以下に説明する形式にしたがってセグメントオプションソースファイルを作成してください。
セグメントオプションソースファイルはオプションリストに記載した内容を参考に、エディタを使用して作成してください。ファイル名は次のとおりとし、YYYの部分はセイコーエプソンより提示される内容を入力してください。

C2XXYYY.SEG

ファイルには、セグメントメモリとSEG端子の対応データおよび出力仕様(セグメント出力/DC・コンプリメンタリ出力/DC・オープンドレイン出力)の選択データを記述します。また、データとは別にコメントの入力も可能です。

以下にセグメントオプションソースファイルの一例を挙げ、記述形式を説明します。

注: 以下の例はコモン出力が4端子に、900H〜AFFHが表示メモリ領域に設定されている機種の場合のものです。機種によりコモン端子数および表示メモリアドレスが異なる場合がありますので、"S5U1C62xxxD Manual"を参照してソースファイルを作成してください。

```
; C2XXYYY.SEG                                    ┐
; LCD SEGMENT DECODE TABLE                       │ コメント
;                                                ┘
 0      901 900 932 A20 S       ;1st DIGIT
 1      912 911 910 923 S
 2      913 920 921 922 S
 3      A00 902 930 931 S
 4      AE0 --- --- --- C       ;DC OUTPUT
 5      AF0 --- --- --- C
 :       :   :   :   :  :
└──────────────────────┘└┘      └──────────┘
   セグメントメモリと    出力仕様の   コメント
   SEG端子の対応データ  選択データ
```

■ コメント

";"以降は行末までがコメントとみなされます。日付やデータの内容、バージョン等が記述できます。

■ セグメントメモリとSEG端子の対応データ

セグメントメモリとSEG端子の対応を示すデータです。オプションリストと配列が同じですので、以下に説明するフォーマットに合わせてオプションリストを見ながらデータを記述してください。

*** セグメントメモリの上位アドレスは機種により異なります。**

＊ SEG端子No.は実際のデバイスに対応しており、その重複は認められません。また、すべてのSEG端子について以下のフォーマットにしたがったデータの記述が必要です。

＊ COMの列の中で使用しないものがある場合は、データとして"---"(ハイフン)を記述します。また、端子そのものを使用しない場合でもSEG端子No.は必要で、COMの欄すべてに "---"(ハイフン)を記述してください。

例: ・SEG8のCOM2を使用しない場合

```
8_981_980_---_A22
```

・SEG12を使用しない場合

```
12_---_---_---_---
```

＊ DC出力を選択する場合はCOM0にセグメントメモリとの対応データを記述し、COM1～COM3には"---"(ハイフン)を記述します。

例: SEG20、SEG21をDC出力にする場合

```
20_933_---_---_---
21_A33_---_---_---
```

＊ "_"はスペースまたはタブを表し、SEG端子No.、COMの各データの間には必ずスペースまたはタブを1つ以上入れます。

■ 出力仕様の選択データ

SEG端子をセグメント出力として使用するか、DC・コンプリメンタリ出力として使用するか、DC・オープンドレイン出力として使用するかの選択データです。
セグメントメモリとSEG端子の対応データの後に、スペースまたはタブを1つ入れてからデータを記述します。

S: セグメント出力
C: DC・コンプリメンタリ出力
P: DC・Pchオープンドレイン出力
N: DC・Nchオープンドレイン出力
] 機種によりどちらか一方が設定されています。

＊ SEG端子の出力仕様選択は2端子ごとの選択となりますので、オプションリストにしたがって注意しながら記述してください。

例: SEG22、SEG23をDC・コンプリメンタリ出力にする場合

```
22_AE0_---_---_--- C
23_AF0_---_---_--- C
```

＊ 使用しないSEG端子はセグメント出力を選択してください。

例: SEG18を使用しない場合

```
18_---_---_---_--- S
```

**注: Evaluation BoardのSEG端子のDC出力はコンプリメンタリのみ対応しています。
そのため、オープンドレインを選択した場合でもコンプリメンタリ出力になります。
必要に応じて外付け回路で対応してください。**

以上説明したフォーマット、制約事項にしたがってセグメントオプションソースファイルを作成してください。

4.2 SOG62XXの起動

SOG62XXを起動させるにはDOSのコマンドレベル(A>などのプロンプトが表示されている状態)で次のプログラム名を入力します。DMS6200から起動させる場合は、メニュー画面で"SOG62XX.EXE"を選択してください。必要に応じてオプションの入力も行います。

```
A>SOG62XX_ [-H]↵
```

_はスペースの入力を表します。
[ ]は省略可能なパラメータであることを表します。
↵はリターンキー(Enter key)の入力を表します。

SOG62XXを起動させる場合は、同じドライブ(ディレクトリ)にセグメントオプションソースファイル(C2XXYYY.SEG)を作成しておく必要があります.

ー H: "ー H"オプションはSOG62XXの入力ファイルとしてセグメントオプションドキュメントファイル(C2XXYYYS.DOC)を用いる場合に指定します.

SOG62XXが起動すると、次の起動メッセージが表示されます。

```
     ***  E0C62XX SEGMENT OPTION GENERATOR. --- Ver 3.00  ***

EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO     NNN      NNN
EEEEEEEEEE   PPPPPPPPPP     SSS    SSSS    OOO    OOO    NNNN     NNN
EEE          PPP     PPP   SSS      SSS   OOO      OOO   NNNNN    NNN
EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN  NNN
EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
EEE          PPP                SSS     OOO      OOO   NNN   NNNNN
EEE          PPP           SSS    SSS    OOO    OOO    NNN    NNNN
EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO     NNN      NN

             (C) COPYRIGHT 1991  SEIKO EPSON CORP.

          SEGMENT OPTION SOURCE FILE NAME IS " C2XXYYY.SEG ".

          THIS SOFTWARE MAKES NEXT FILES.

             C2XXYYYS.HEX  ...  SEGMENT OPTION HEX FILE.
             C2XXYYYS.DOC  ...  SEGMENT OPTION DOCUMENT FILE.

                        STRIKE ANY KEY.
```

"STRIKE ANY KEY"が表示されたところで任意のキー入力を行い、プログラムを先に進めます。
中止する場合はCTRL+C(使用するPCによりSTOPも可)を入力してください。DOSのコマンドレベルに戻ります。

起動メッセージの後、画面には現在パソコンに設定されている日付が表示され、新たな日付の入力を要求します。

```
*** E0C62XX USER'S OPTION SETTING. --- Ver 3.00 ***

CURRENT DATE IS  91/07/19
PLEASE INPUT NEW DATE  :   91/07/22↵
```

日付を変更する場合は年月日をそれぞれ2桁ずつ、"/"で区切って入力します。
日付を変更しない場合は"↵"キーのみを入力して、先に進めてください。

4.3 入力ファイルの選択

以下に入力ファイルの選択方法を示します。

```
*** SOURCE FILE(S) ***

C2XX0A0        C2XX0B0        C2XX0B1        C2XX0C0
```

日付入力の後、カレントドライブ上の入力可能なファイルの一覧が表示されます。

- **Hオプションを指定しない場合**
  カレントドライブ上のセグメントオプションソースファイルの一覧を表示します。
  ソースファイルが存在しない場合は次のメッセージを表示してプログラムを終了します。

```
SEGMENT OPTION SOURCE FILE IS NOT FOUND.
```

- **Hオプションを指定した場合**
  カレントドライブ上のセグメントオプションドキュメントファイルの一覧を表示します。
  ドキュメントファイルが存在しない場合は次のメッセージを表示してプログラムを終了します。

```
SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.
```

```
*** SOURCE FILE(S) ***

C2XX0A0       C2XX0B0       C2XX0B1       C2XX0C0

 PLEASE INPUT SEGMENT SOURCE FILE NAME? C2XX0A0↵                          ... (1)
 PLEASE INPUT USER'S NAME? SEIKO EPSON CORP.↵                             ... (2)
 PLEASE INPUT ANY COMMENT
 (ONE LINE IS 50 CHR)? TOKYO DESIGN CENTER↵                               ... (3)
                     ? 421-8 HINO HINO-SHI TOKYO 191 JAPAN↵
                     ? ↵
```

(1) PLEASE INPUT SEGMENT SOURCE FILE NAME?

- **Hオプションを指定しない場合**
  セグメントオプションソースファイル名を入力します(エクステンションは入力しないでください)。
  カレントドライブ上に指定した名前のファイルが存在しない場合、次のようにエラーメッセージが表示され、再度ファイル名の入力状態になります。

  例: `PLEASE INPUT SEGMENT SOURCE FILE NAME? C2XX0N0↵`
  `SEGMENT OPTION SOURCE FILE IS NOT FOUND.`

- **Hオプションを指定した場合**
  セグメントオプションドキュメントファイル名を入力します(エクステンションは入力しないでください)。
  カレントドライブ上に指定した名前のファイルが存在しない場合、次のようにエラーメッセージが表示され、再度ファイル名の入力状態になります。

  例: `PLEASE INPUT SEGMENT DOCUMENT FILE NAME? C2XX0N0↵`
  `SEGMENT OPTION DOCUMENT FILE IS NOT FOUND.`

(2) PLEASE INPUT USER'S NAME?

お客さまの会社名を入力します。

(3) PLEASE INPUT ANY COMMENT

コメントを入力します。
1行に入力可能な文字数は50文字まで(それ以上の入力は50文字以降が削除されます)で、最大10行まで入力することができます。コメントの入力を終了する場合は必要な行数の入力を行い、次の行頭でリターンキーを入力します。

コメントには次のような内容を含めるようお願いします。

・事業所、所属
・所在地、電話番号、FAX番号
・その他、技術情報等

以上の入力が終了するとHEXファイルの作成とEPROMの選択作業に入ります。

4.4 HEXファイルの作成とEPROMの選択

```
END OF OPTION SETTING.
 DO YOU MAKE HEX FILE (Y/N) ? Y↵                    ... (1)

 *** OPTION EPROM SELECT MENU ***

        1. 27C64
        2. 27C128
        3. 27C256
        4. 27C512

 PLEASE SELECT NO.? 2↵                              ... (2)

        2. 27C128   SELECTED
```

入力ファイルの設定が終了するとHEXファイル作成の有無について問い合わせてきます。Evaluation Boardを使用してプログラムのデバッグを行う場合、HEXファイル(C2XXYYYS.HEX)が必要となります。

(1) DO YOU MAKE HEX FILE (Y/N) ?

HEXファイル(C2XXYYYS.HEX)を作成する場合は"Y"を入力します。"N"を入力した場合はHEXファイルは作成されず、ドキュメントファイル(C2XXYYYS.DOC)のみが作成されます。
ただし、Hオプションが指定された場合は無条件にHEXファイルが作成されます。よって、本メッセージは表示されません。

(2) PLEASE SELECT NO.?

(1) で"Y"を入力した場合、オプションROM選択メニューが表示されますので、Evaluation Boardのオプション設定に使用するEPROMを選択します。(1) で"N"を入力した場合、このメニューは表示されません。
上記の例では"27C128"を選択しています。

```
MAKING FILE IS COMPLETED.
```

ここまでの操作が終了するとSOG62XXは各ファイルを作成します。セグメントオプション設定にエラーが存在しなければ左のメッセージが表示され、SOG62XXを終了します。

注: Evaluation Boardに装着するEPROMは以下の規格のものをご使用ください。

TACC ≦ 170nsec (TACC: アクセスタイム)

4.5 終了方法

一連の操作が終了するとSOG62XXは終了します。

他の方法として、プログラム実行中にCTRL+C(使用するPCによりSTOPも可)を入力することにより強制的に終了させることができます。

5 エラーメッセージ

セグメントオプションソースファイルにエラーが存在する場合、次のようなメッセージが表示されます。この場合には、セグメントオプションHEXファイルは作成されません。また、セグメントオプションドキュメントファイルはセグメントオプションソースファイルにエラーメッセージを付加したものになります。

注: 以下の例はコモン出力が4端子、セグメント出力が26端子、900H～AFFHが表示メモリ領域に設定されている機種の場合のものです。機種により出力ポート数および表示メモリアドレスが異なる場合がありますので、注意してください。

```
N 12   66      9B0        9B1        9B2        9B3        S
S 16   15      9F0MSD     9F1        9F2        9F3        S
D 20   19      A30        A31        A32        A31        S
N 22   42      A50        A51        A52        A53        S
D 23   22      A60        A61        A31        A31        S
R 25   24      A80        881        A82        A83        S

Duplication is  SEG NO. 19  COM NO. 3
Duplication is  SEG NO. 22  COM NO. 2
Duplication is  SEG NO. 22  COM NO. 3

7 ERROR(S)

STRIKE ANY KEY.

MAKING SEGMENT OPTION FILES IS NOT COMPLETED BY SOURCE FILE ERROR-(S).
```

エラーが発生した場合、上例のように0カラム目のエラーシンボルに続きエラーの発生したソースリストが表示されます。SOG62XXには以下の4種類のエラーシンボルが設定されています。

- **S … Syntax Error**
- **N … Segment No. Select Error**
- **R … RAM Address Select Error**
- **D … Duplication Error**

エラーの優先順位はS・N・R・Dの順です。

以下、各エラーについて解説します。

[S] Syntax Error

データの記述がフォーマットどおりになっていない場合に発生します。
フォーマットどおりの記述になるようセグメントオプションソースファイルを修正してください。

例:
```
S 16  15     9F0MSD  9F1     9F2     9F3     S
             ↑
```
この部分の記述がフォーマットどおりになっていない。

[N] Segment No. Select Error

セグメントNo.の記述が指定可能な範囲を越えた場合に発生します。
設定されているセグメントNo.の範囲内になるようセグメントオプションソースファイルを修正してください。

例:
```
N 12  66     9B0     9B1     9B2     9B3     S
N 22  42     A50     A51     A52     A53     S
      ↑
```
この部分の記述が指定可能な範囲を越えている。

[R] RAM Address Select Error

セグメントメモリのアドレスおよびデータビットの記述が指定範囲外の場合に発生します。
アドレスが表示メモリの範囲内、データビットが"0～3"の値になるようセグメントオプションソースファイルを修正してください。

例:
```
R 25  24     A80     881     A82     A83     S
             ↑
```
この部分の記述が指定可能な範囲外。

[D] Duplication Error

記述したデータ(SEG端子No.、セグメントメモリのアドレスおよびデータビット)が重複している場合に発生します。
データが重複しないようセグメントオプションソースファイルを修正してください。

例:
```
D 20  19     A30     A31     A32     A31     S

D 23  22     A60     A61     A31     A31     S
                                                "A31"が重複

Duplication is SEG NO. 19 COM NO. 3
Duplication is SEG NO. 22 COM NO. 3
```

"Duplication is ..."のメッセージは後から重複したもののみ表示されます。

以上のエラーシンボル以外に、次のエラーメッセージが表示される場合があります。

Out Port Set Error

出力仕様の記述が2端子ごとになっていない場合に発生します。
2端子ごとの指定になるようセグメントオプションソースファイルを修正してください。

例: `Segment No. 18 - 19 Out Port Set Error`

前記4つのエラーのいずれかが発生した場合、このエラーのチェックは行われません。そのため前記のエラーを修正した後に、発生する場合があります。

エラーが発生した場合、表示されるメッセージはセグメントオプションドキュメントファイルを参照することにより確認できます。オプションリストと対比しながらセグメントオプションソースファイルを修正し、再度プログラムを実行してください。

次にエラー発生時のセグメントオプションドキュメントファイル例を示します。

```
 LINE  SOURCE STATEMENT

   1   0       900     901     902     903     S
   2   1       910     911     912     913     S
   3   2       920     921     922     923     S
   4   3       930     931     932     933     S
   5   4       940     941     942     943     S
   6   5       950     951     952     953     S
   7   6       960     961     962     963     S
   8   7       970     971     972     973     S
   9   8       980     981     982     983     S
  10   9       990     991     992     993     S
  11   10      9A0     9A1     9A2     9A3     S
N 12   66      9B0     9B1     9B2     9B3     S
  13   12      9C0     9C1     9C2     9C3     S
  14   13      9D0     9D1     9D2     9D3     S
  15   14      9E0     9E1     9E2     9E3     S
S 16   15      9F0MSD  9F1     9F2     9F3     S
  17   16      A00     A01     A02     A03     S
  18   17      A10     A11     A12     A13     S
  19   18      A20     A21     A22     A23     S
D 20   19      A30     A31     A32     A31     S
  21   20      A40     A41     A42     A43     S
N 22   42      A50     A51     A52     A53     S
D 23   22      A60     A61     A31     A31     S
  24   23      A70     A71     A72     A73     S
R 25   24      A80     881     A82     A83     S
  26   25      A90     A91     A92     A93     S

  S --- Syntax Error
  N --- Segment No. Select Error
  R --- RAM Address Select Error
  D --- Duplication Error

Duplication is   SEG NO. 19  COM NO. 3
Duplication is   SEG NO. 22  COM NO. 3
```

VII

Evaluation Board

S5U1C62xxxE

ここでは4-bit Single Chip Microcomputer S1C62XXX用Evaluation Board S5U1C62xxxEの機能と操作方法を中心に解説を行っています。

Evaluation Board S5U1C62xxxE

- 目 次 -

1 機種による相違点 VII-1

2 S5U1C62xxxEの概要 VII-1

3 使用上の注意 VII-1

3.1 操作上の注意事項 VII-1

3.2 実際のICとの相違点 VII-2

4 各部の名称と機能 VII-2

4.1 基本機能 VII-2

4.2 操作パネル(上面) VII-3

4.3 上面カバー内部 VII-4

4.4 前面パネル VII-5

4.5 後面パネル VII-5

4.6 底面カバー内部 VII-5

5 ケーブルの接続 VII-6

5.1 ICE(S5U1C62000H)との接続 VII-6

5.2 電源ケーブルの接続 VII-6

5.3 ターゲットシステムとの接続 VII-6

6 S5U1C62xxxEの操作方法 VII-7

6.1 準備作業 VII-7

6.1.1 ターゲットシステムの作成 VII-7

6.1.2 ROMの作成/装着 VII-7

6.2 S5U1C62xxxEの単独使用 VII-8

6.2.1 電源のON/OFF VII-8

6.2.2 デバッグ操作 VII-8

6.3 ICE(S5U1C62000H)接続時の操作 VII-9

6.3.1 電源のON/OFF VII-9

6.3.2 デバッグ操作 VII-9

6.4 動作試験 VII-9

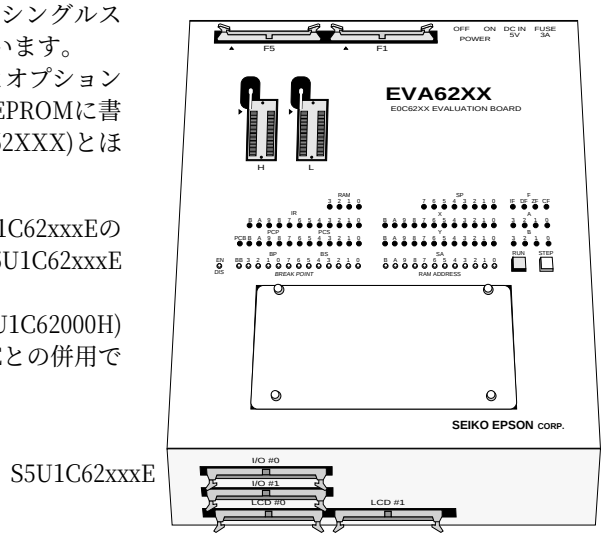
1 機種による相違点

S5U1C62xxxEは実際のIC(S1C62XXX)と同等の機能を持ちます。操作方法等は同様ですが、I/OおよびLCDコネクタの端子配置と入出力信号の仕様が異なります。また、上面パネル内の配置等も異なります。
各機種の詳細についてはハードウェアに添付の"S5U1C62xxxE Manual"を参照してください。

2 S5U1C62xxxEの概要

S5U1C62xxxEはS1C62XXXの開発ツールで、シングルステップ、プログラムブレーク機能等を持っています。
S5U1C62xxxEにアプリケーションプログラムとオプションジェネレータで作成したオプションデータをEPROMに書き込み実装することにより、実際のCPU(S1C62XXX)とほぼ同等の機能を実現することができます。

デバッグ機能の操作やCPU内部のモニタはS5U1C62xxxEの操作スイッチおよびLED表示で行えるため、S5U1C62xxxE単体によるデバッグ作業が可能です。
さらに、インサーキットエミュレータICE(S5U1C62000H)とのインターフェース機能を持っており、ICEとの併用でより高度なデバッグが可能となります。



S5U1C62xxxE

※開発ツール本体には、旧製品名(EVA62XX)が記載されております。

3 使用上の注意

S5U1C62xxxEを正しく使用していただくために、以下の事項に注意してください。

3.1 操作上の注意事項

- ケーブルの接続、切り離しは接続する機器すべての電源をOFFにした状態で行ってください。
- S5U1C62xxxEのPOWERスイッチをOFFにした後、再びONにする場合は、その間に10秒以上の時間をおいてください。
- ROMソケットL、HにROMを挿入した場合、レバーは水平に倒して確実にロックしてください。
  ROMを取り外した場合もレバーを水平にロックしておいてください。レバーを立てた状態で長く放置しますと接触不良が発生しやすくなります。
- ROMが正しく装着されていることを確認してからS5U1C62xxxEを動作させてください。
- S5U1C62xxxEが正しく動作しない場合は、動作試験(S5U1C62xxxE Manual参照)を行ってください。

3.2 実際のICとの相違点

実際のICとは機能上以下の相違がありますので、使用に際してご注意ください。

■ I/Oについての相違

ロジックレベル(S5U1C62xxxEは5V)および出力ドライブ能力、プルダウン/アップ抵抗値の差により応答時間が変わります。特にキースキャンルーチンを作成する場合は、キーOFF時のプルダウン/アップ抵抗による応答時間に注意してください。

■ LCDについての相違

- コントラストを調整するVADJボリュームが付いています。ただし実際のICはレベルが固定されていて調整できません。
- PchまたはNchオープンドレンのオプションは設定できません。
- 出力ドライブ能力が異なります。

■ POWER ONでのシーケンスの相違

S5U1C62xxxEでは、POWER ONによりコンフィグレーションを行い内部状態を確定後、実際のICと同等の機能となります。このため、コンフィグレーション終了まで入出力状態が不確定となり、POWER ONリセットの時間等に影響がでます。

■ 機能上の相違

ロジックレベルが高いため、発振開始や停止時間などのタイミングが実際のICと異なります。

機種により機能が異なりますので、他の相違点については"S5U1C62xxxE Manual"を参照してください。

4 各部の名称と機能

ここでは各部の名称と機能について説明します。

4.1 基本機能

以下にS5U1C62xxxEの基本機能を示します。

■ プログラムの実行 (プログラムRUN機能)

EPROMに書き込んだアプリケーションプログラムをセットすることにより、そのプログラムを実行します。

■ プログラムの1ステップ動作 (シングルステップ機能)

プログラムを1ステップずつ実行することができ、各インストラクションごとに変わるCPUの内部状態が確認できます。

■ 任意のアドレスでプログラム実行停止 (プログラムブレーク機能)

任意のアドレスにブレークポイントを設定でき、プログラムの実行をそこで中断できます。
ブレークポイント以降のプログラム実行はプログラムRUN機能で再開できます。

■ ブレーク中のプログラムアドレスおよび命令コードのモニタ

LED表示により、プログラムアドレスと命令コードが確認できます。

■ ブレーク中のRAM、各レジスタおよびフラグ等のモニタ

LED表示により、RAM、A/B/X/Yレジスタ、スタックポインタ、フラグの内容が確認できます。

■ ICEとのインタフェース

ICEとの接続が可能で、より高度なデバッグ環境が得られます。

■ オプションROMの装着により、ハードウェアオプションを設定

ファンクションオプションジェネレータで作成したファンクションおよびセグメントオプションジェネレータで作成したセグメントのオプションデータをEPROMに書き込み、装着することにより、ハードウェアオプションであるI/Oポートやセグメントの仕様を設定できます。

4.2 操作パネル (上面)

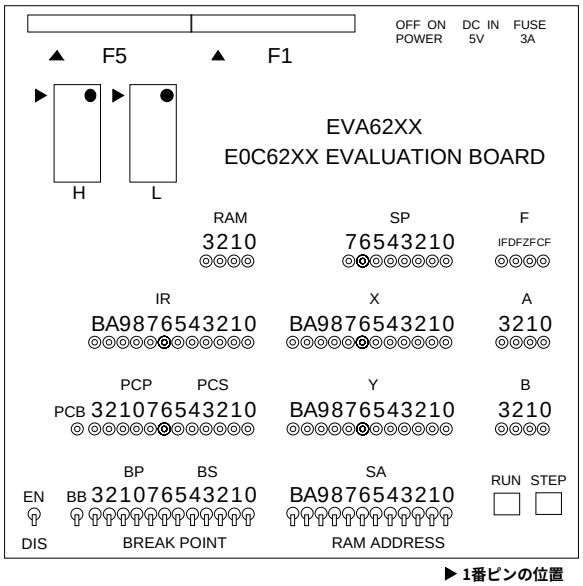


図4.2.1 操作パネル

■ 操作スイッチ、キー

- **EN/DIS スイッチ**
  ブレークポイントの設定を有効/無効にします。スイッチがEN(Enable)側でブレークポイントの設定は有効となり、DIS(Disable)側で無効となります。通常はDIS側に倒しておいてください。

- **BREAK POINT スイッチ (BB、BP、BS)**
  プログラムの実行を停止させるブレークポイントアドレスを設定します。BB、BP、BSはそれぞれブレークポイントアドレスのバンク、ページ、ステップを設定するスイッチで、上側に倒すと"1"、下側で"0"になります。
  BREAK POINTスイッチで設定したブレークポイントアドレスはEN/DISスイッチをEN側に倒した場合に有効で、設定したアドレスと実行中のプログラムアドレスが一致するとプログラムブレーク状態となり、そのアドレスの命令を実行する直前でプログラムを停止します。EN/DISスイッチをDIS側に倒した場合、この機能は働きません。

- **RAM ADDRESS スイッチ (SA)**
  プログラムブレーク時にRAMの内容を確認するためのスイッチで、RAMのアドレスを設定します。上に倒すと"1"、下に倒すと"0"が設定されます。このスイッチで設定されたアドレスの内容はRAM表示LEDに表示されます。

- **RUN キー**
  プログラムブレーク状態からプログラムを再スタートさせるキーで、このキーの入力によりブレークアドレス以降のプログラムが実行されます。

- **STEP キー**
  プログラム実行中にこのキーを入力すると、その時点でプログラムブレーク状態になります。
  プログラムブレーク状態で入力した場合はブレークしたアドレスの命令を1ステップ実行し、次のアドレスで再びプログラムブレーク状態になります。これにより、プログラムを1ステップずつ実行させることができます。

■ LED

CPUの内部状態がLED表示されます。
LEDは点灯が"1"、消灯が"0"のバイナリ表示が行われます。

- **RAM (3210)** RAM ADDRESSスイッチで設定されたRAMアドレスの内容が表示されます。
- **IR (BA9876543210)** 現在のアドレスの命令コードが表示されます。ブレークポイントで停止している場合、表示されている命令は実行前の状態です。
- **PCB** バンクアドレスが表示されます。
- **PCP (3210)** ページアドレスが表示されます。
- **PCS (76543210)** ステップアドレスが表示されます。
- **SP (76543210)** スタックポインタの値が表示されます。
- **X (BA9876543210)** インデックスレジスタXの内容が表示されます。

- **Y (BA9876543210)** インデックスレジスタYの内容が表示されます。
- **F/IF** インタラプトフラグの状態が表示されます。
- **F/DF** デシマルフラグの状態が表示されます。
- **F/ZF** ゼロフラグの状態が表示されます。
- **F/CF** キャリーフラグの状態が表示されます。
- **A (3210)** Aレジスタの内容が表示されます。
- **B (3210)** Bレジスタの内容が表示されます。

■ ROMソケット

- **L (low)、H (high)**
  ターゲットプログラムROM用のICソケットです。
  Lには命令コードの下位8ビット(I7～I0)のROM(L.HEX)を、Hには上位4ビット(IB～I8)のROM(H.HEX)を装着します。動作試験時には、診断用ROMを挿入します。

■ コネクタ

- **F1、F5**
  ICEインタフェースケーブルを接続するためのコネクタです。

4.3 上面カバー内部

上面カバーの内部は機種により配置および内容が異なります。
基本的には以下の内容がここに配置されます。詳細は"S5U1C62xxxE Manual"を参照してください。

- **RESET スイッチ**
  CPUをリセットし、ターゲットプログラムをPAGE=01H、STEP=00Hからスタートさせます。

- **VADJ**
  LCDのコントラストを調整するボリュームです。("S5U1C62xxxE Manual 2.2 実際のICとの相違点"を参照してください。)
  LCDドライバを持たない機種には設定されません。

- **VSVD**
  電源電圧検出機能(SVD)を確認するため、擬似的に電源電圧を変化させるボリュームです。
  ("S5U1C62xxxE Manual 2.2 実際のICとの相違点"を参照してください。)
  SVD機能を持たない機種には設定されません。

- **DONE LED**
  S5U1C62xxxEが電源ONによりコンフィグレーションを終了し、デバッグが可能な状態になるとLEDが点灯します。電源ON後数秒経過しても点灯しない場合は、再度電源を入れ直してください。

- **ROMソケット (F.HEX)**
  ファンクションオプションジェネレータ(FOG62XX)によって作成したファンクションオプション設定用のROM(F.HEX)を装着するICソケットです。

- **LEDおよびCHK端子**
  特定のI/Oレジスタの値("1"または"0")を示すLEDとオシロスコープ等による確認用の端子が配置されます。

4.4 前面パネル

前面パネルにはターゲットシステム接続用のコネクタがあります。機種により構成が異なります。

- **I/O #0、I/O #1**
  I/O接続用コネクタです。I/Oケーブルによりターゲットシステムと接続します。
- **LCD #0、LCD #1**
  LCD接続用コネクタです。LCDケーブルによりターゲットシステムと接続します。

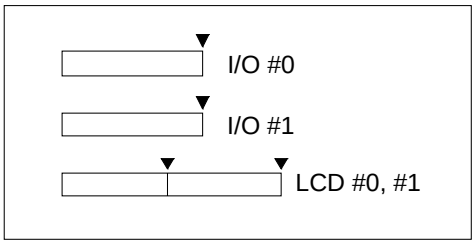


図4.4.1　前面パネル

4.5 後面パネル

後面パネルには外部電源入力部があります。

- **POWERスイッチ (ON/OFF)**
  S5U1C62xxxEへの外部電源の供給をON/OFFするスイッチです。
  ICE接続時にはOFFにします。
- **FUSE**
  外部電源用の管形ヒューズで、3A以上の電流で熔断します。
- **DC IN 5V**
  外部電源接続コネクタです。
  S5U1C62xxxEを単独使用する際にはDC5V、容量3A以上の安定化電源を使用してください。
  ICEを接続して使用する場合、電源はICEから供給されますので、外部電源は必ず外して使用してください。

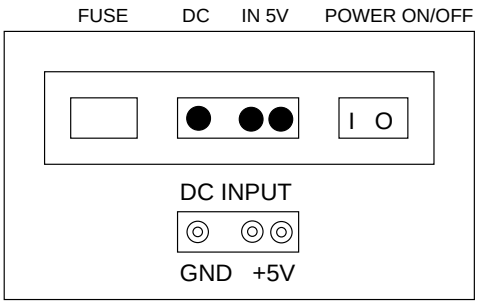


図4.5.1　後面パネル

4.6 底面カバー内部

- **ROMソケット**
  セグメントオプションジェネレータ(SOG62XX)によって作成したLCDセグメント割り付け用のROM(S.HEX)を挿入するICソケットで、▶印が1番ピンを示します。
  同じ内容のROMを2個作成し、このソケットに装着します。

  セグメントオプションを持たない機種には、このソケットがありません。

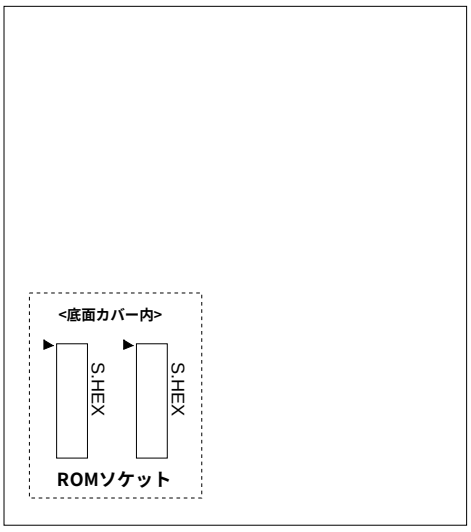


図4.6.1　底面カバー内部

5 ケーブルの接続

ここではS5U1C62xxxEと電源ケーブル、ICEおよびターゲットシステムとの接続について説明します。

注: ケーブルの接続、切り離しは接続する機器すべての電源をOFFにして行ってください。

5.1 ICE(S5U1C62000H)との接続

S5U1C62xxxEとICEはそれぞれのF1コネクタ、F5コネクタ間を、ICEに添付のインタフェースケーブル(2本)で接続します。
S5U1C62xxxEのF1、F5コネクタにはコネクタの誤挿入防止突起が外側にある方を接続します。
ICEのF1、F5コネクタにはコネクタの誤挿入防止突起が内側(ケーブル側)にある方を接続します。
図5.1.1と図5.1.2にICEインタフェースケーブル外観図と接続図を示します。

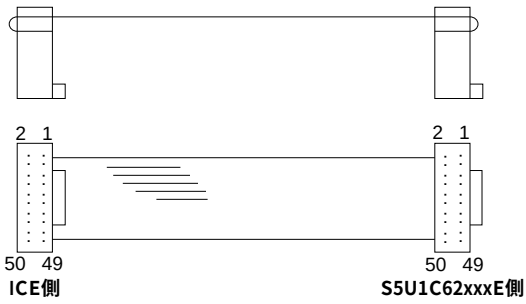


図5.1.1 ICEインタフェースケーブル外観図

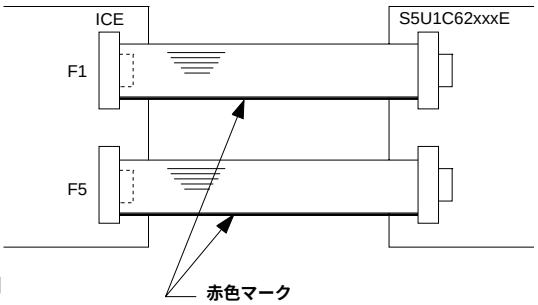


図5.1.2 ケーブル接続図

注: S5U1C62xxxEには外部電源入力コネクタ+5V(V_{DD})、GND(V_{SS})がありますが、ICEと接続する場合には、この端子には何も接続しないでください。

5.2 電源ケーブルの接続

S5U1C62xxxEを単独で使用する場合には、電源ケーブルを接続して外部よりDC5V(3A以上)を供給します。
ICEを接続する場合には、電源がICEから供給されますので電源ケーブルは必要ありません。接続されている場合には取り外してください。
図5.2.1に電源ケーブルピン接続図を示します。

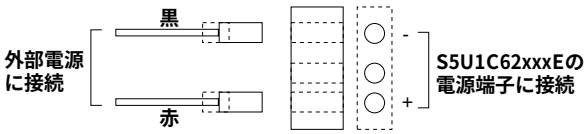


図5.2.1 電源ケーブルピン接続図

5.3 ターゲットシステムとの接続

S5U1C62xxxEとターゲットシステムとの接続は、I/O #0、I/O #1、LCD #0、LCD #1コネクタを使用して行います。

LCD #0コネクタとLCD #1コネクタから出力される信号はファンクションレベルで実際のICと同等ですので、そのままターゲットシステムのLCDと接続することができます。
LCDコントラスト(LCD駆動電圧)はVADJボリュームで調整できます。(実際のICは駆動電圧固定)

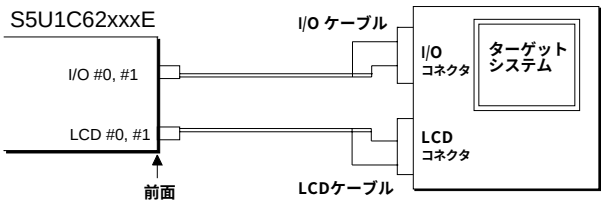


図5.3.1 ターゲットシステムの接続

コネクタの構成、ピン配置については"S5U1C62xxxE Manual"を参照してください。

6 S5U1C62xxxEの操作方法

6.1 準備作業

ここではS5U1C62xxxEの単独使用とICEに接続する場合に共通する準備作業について説明します。接続方法については"5 ケーブルの接続"を参照してください。
なお、S5U1C62xxxEの動作確認については付属の診断用ROMを装着して、動作試験を行ってください。この動作試験は定期的に行うことをお勧めします。("S5U1C62xxxE Manual"参照)
以下の作業はS5U1C62xxxEのPOWER SWをOFFにして行ってください。

6.1.1 ターゲットシステムの作成

LCDパネル/キー/スイッチ等を基板上に実装し、ターゲットシステムを作成します。
S5U1C62xxxEとの接続端子としてはS5U1C62xxxEに添付のI/Oコネクタおよび LCDコネクタを使用してください。(各コネクタのピン配列については、"S5U1C62xxxE Manual"を参照してください。)

注: S5U1C62xxxEは仕様上で実際のCPUと若干の相違点がありますので、ターゲットシステムの作成は"S5U1C62xxxE Manual 2.2 実際のICとの相違点"を参照して行ってください。

6.1.2 ROMの作成/装着

プログラムROM、ファンクションオプションROM、セグメントオプションROMを作成し、S5U1C62xxxEの各ソケットに装着します。S5U1C62xxxE出荷時には診断用としてファンクションオプションROM、セグメントオプションROMが装着されていますので、これを取り外して作成したROMを装着してください。

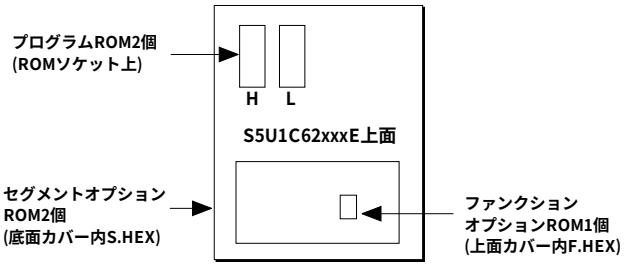


図6.1.2.1 ROMの装着

- **プログラムROM (2個)**
  プログラムROMはアプリケーションプログラムの命令コードが書き込まれたROMのことです。
  クロスアセンブラASM62XXで出力したオブジェクトファイルをEPROMに書き込み、プログラムROMを作成します。オブジェクトファイルは命令コードの上位(C2XXYYYH.HEX)、下位(C2XXYYYL.HEX)の2つが出力されますのでROMも2個作成し、H.HEXを上面パネルのソケットHに、L.HEXをソケットLに装着します。ICE接続時にはこのROMは必要ありません。

- **ファンクションオプションROM (1個)**
  ファンクションオプションROMはI/Oポート仕様等のファンクションオプションを設定するROMです。
  ファンクションオプションジェネレータで出力したファンクションオプションHEXファイル(C2XXYYYF.HEX)からファンクションオプションROMを作成し、上面カバー内のROMソケット(F.HEX)に装着します。

- **セグメントオプションROM (2個) ...セグメントオプションの設定された機種のみ**
  セグメントオプションROMはセグメント出力ポートの仕様を設定するROMです。
  セグメントオプションジェネレータで出力したセグメントオプションHEXファイル(C2XXYYYS.HEX)からセグメントオプションROMを2個(同じ内容)作成し、底面カバー内のソケット(S.HEX)に装着します。

- **EPROMの仕様**
  EPROMは以下の仕様のものを使用してください。

プログラムROM:27C64～27C512	アクセスタイム	250ns 以下
オプションROM:27C64～27C512	アクセスタイム	250ns 以下
セグメントROM:27C64～27C512	アクセスタイム	170ns 以下

6.2 S5U1C62xxxEの単独使用

ここではS5U1C62xxxEを単独に使用する場合の操作について説明します。
S5U1C62xxxEは外部電源を接続することにより単独使用が可能です。
外部電源としてはDC5V安定化電源(3A以上)を使用し、極性(±)を間違えないように接続してください。
("5.2 電源ケーブルの接続"参照)

6.2.1 電源のON/OFF

S5U1C62xxxEの電源をONにする前には必ず以下のことを確認してください。

(1) 電源ケーブルが正しく接続されていること
(2) ターゲットシステムが正しく接続されていること
(3) ROMが正しく装着されていること

以上を確認のうえ、次の順序で電源をONにします。電源をOFFにする場合は逆の順序で行ってください。

(1) 安定化電源をON
電圧可変タイプの場合は出力電圧を5Vに合わせてください。
(2) S5U1C62xxxEのPOWER SWをON

注: S5U1C62xxxEのPOWER SWをOFFにして再びONにする場合は、その間に10秒以上の時間をおいてください。

S5U1C62xxxEのPOWER SWをONにすると、数秒以内に上面カバー内のDONE LED(グリーン)が点灯してデバッグ作業が開始できることを示します。
POWER SWのONから10秒以上経過してもDONE LEDが点灯しない場合は次の操作を行ってください。

(1) S5U1C62xxxEのPOWER SWをOFF
(2) ROMの装着とケーブルの接続を確認
(3) ヒューズの確認
(4) S5U1C62xxxEのPOWER SWをON

以上の操作を行ってもDONE LEDが点灯しない場合は一度自己診断を行ってください。
自己診断の方法については"S5U1C62xxxE Manual"を参照してください。

6.2.2 デバッグ操作

S5U1C62xxxEの単独使用におけるデバッグ機能と操作方法は以下のとおりです。

- **プログラムフリーラン**
  RESET SW(上面カバー内)を押すことによりS5U1C62xxxEはプログラムフリーランの状態になり、アプリケーションプログラムをPAGE1、STEP0から実行します。S5U1C62xxxEの電源ON直後にRESET SWを押す場合は、DONE LEDが点灯したことを確認してから行ってください。

- **プログラムブレーク**
  BREAK POINTスイッチによって設定したアドレスでプログラムを停止させる機能です。
  この機能はEN/DISスイッチをEN側にセットした場合に有効となり、実行中のプログラムアドレスが設定されたブレークポイントに一致したところでプログラムを停止します。なお、ブレークポイントにある命令の実行前にプログラムは停止します。STEPキーの入力により、その時点でプログラムを停止させることもできます。プログラムが停止するとCPUの内部状態を示すLEDインジケータも現在の状態を表示しますので、この内容をプログラムと照らし合わせてデバッグを行ってください。
  プログラムブレーク状態から実行を再開するには、次のブレークポイントを設定してRUNキーを押します。なお、RUNキーの代わりにSTEPキーを押すことにより、次に説明するシングルステップ動作を行わせることもできます。

- **シングルステップ**
  プログラムブレーク後STEPキーを押すことにより、現在のアドレスの命令1ステップのみを実行し、次のアドレスで再びプログラムを停止(プログラムブレーク状態)します。
  この機能によりプログラムの実行状態が確認できます。

その他の機能については"S5U1C62xxxE Manual"を参照してください。

6.3 ICE(S5U1C62000H)接続時の操作

ここではS5U1C62xxxEをICEに接続して使用する場合の操作について説明します。
ICE接続時にはS5U1C62xxxEを以下のように設定してください。

(1) 外部電源は接続しない
(2) POWER SWをOFFにしておく
(3) 操作パネルのスイッチをすべて下向きにセット

6.3.1 電源のON/OFF

S5U1C62xxxEの電源はICEより供給され、電源のON/OFFはICEのPOWER SWにより行います。
S5U1C62xxxEのPOWER SWはOFFのままにしておきます。

注: ICEのPOWER SWをOFFにして再びONにする場合は、その間に10秒以上の時間をおいてください。

ICEのPOWER SWをONにすると、数秒以内にS5U1C62xxxEの上面カバー内にあるDONE LED(グリーン)が点灯してデバッグ作業が開始できることを示します。
POWER SWのONから10秒以上経過してもDONE LEDが点灯しない場合は次の操作を行ってください。

(1) ICEのPOWER SWをOFF
(2) ICEのサーキットブレーカONを確認
(3) ROMの装着とケーブルの接続を確認
(4) ICEのPOWER SWをON

以上の操作を行ってもDONE LEDが点灯しない場合は、一度S5U1C62xxxEの自己診断を行ってください。
自己診断の方法については"S5U1C62xxxE Manual"を参照してください。

6.3.2 デバッグ操作

デバッグ操作はホストコンピュータ上で行い、S5U1C62xxxEの制御はICE側から行われます。
操作方法については"VIII ICE Control Software ICS62XX"を参照してください。

S5U1C62xxxEのRESET SW以外のスイッチ、LED表示は無効となります。S5U1C62xxxE側のスイッチ操作は行わないでください。
また、ターゲットプログラムROMが装着されていても機能しません。

6.4 動作試験

S5U1C62xxxEには自己診断機能があります。
動作試験を行う場合には、出荷時に装着されているファンクションオプションROM、セグメントオプションROM、添付品の診断用プログラムROMが必要です。これらのROMが取り外されている場合は、各ソケットに装着し直してください。
また、S5U1C62xxxEを単独使用するため、外部電源(DC5V、3A)も接続します。
動作試験の詳細については"S5U1C62xxxE Manual"を参照してください。

VIII

ICE Control Software

ICS62XX

本書では4-bit Single Chip Microcomputer S1C62XXX用のソフトウェア開発支援装置S5U1C62000Hの機能、およびICEコントロールソフトウェアICS62XXの操作方法を解説します。

ICE Control Software ICS62XX

- 目 次 -

1 機種による相違点 VIII-1

2 S5U1C62000H仕様 VIII-2

2.1 概要 VIII-2
2.1.1 特長 VIII-2
2.1.2 ソフトウェア構成 VIII-2
2.1.3 機能一覧表 VIII-3
2.1.4 機能別コマンド一覧表 VIII-4
2.1.5 アルファベット順コマンド一覧表 VIII-6

2.2 システムの接続と起動 VIII-8
2.2.1 ホストシステムの設定 VIII-8
2.2.2 ICS62XXの起動方法 VIII-9

2.3 S5U1C62000Hの動作と機能 VIII-10
2.3.1 動作概要 VIII-10
2.3.2 ブレークモードとブレーク機能 VIII-11
2.3.3 SYNC端子、HALT端子出力 VIII-12
2.3.4 ラン中の表示とブレーク時の表示 VIII-13
2.3.5 ブレーク指定コマンド VIII-13
2.3.6 ターゲット割り込みとブレーク VIII-14
2.3.7 ヒストリ機能 VIII-14
2.3.8 ブレーク遅延機能 VIII-16
2.3.9 カバレッジ機能 VIII-16
2.3.10 実行時間の測定 VIII-16
2.3.11 自己診断機能 VIII-16
2.3.12 プリンタの起動 VIII-17
2.3.13 エミュレーション中の制限事項 VIII-17

2.4 コマンドの詳細 VIII-18
2.4.1 DISPLAYコマンド群 VIII-19
• Lコマンド VIII-20
• DPコマンド VIII-22
• DDコマンド VIII-24
• DRコマンド VIII-26
• Hコマンド VIII-27
• HB,HGコマンド VIII-30
• HS,HSR,HSWコマンド VIII-32
• HP,HPS,CHKコマンド VIII-33
• DXY,CVD,CVRコマンド VIII-34
2.4.2 SETコマンド群 VIII-35
• Aコマンド VIII-36
• FPコマンド VIII-37
• FDコマンド VIII-38
• MPコマンド VIII-39
• MDコマンド VIII-40
• SPコマンド VIII-41
• SDコマンド VIII-42
• SRコマンド VIII-43
• SXY,HCコマンド VIII-44
• HA,HAD,HARコマンド VIII-45

2.4.3 BREAK & GOコマンド群 VIII-47
• BA,BARコマンド *VIII-48*
• BD,BDRコマンド *VIII-49*
• BR,BRRコマンド *VIII-50*
• BM,BMRコマンド *VIII-52*
• BCコマンド *VIII-54*
• BRESコマンド *VIII-55*
• Gコマンド *VIII-56*
• Tコマンド *VIII-58*
• Uコマンド *VIII-60*
• BE,BSYN,BTコマンド *VIII-61*
• BRKSELコマンド *VIII-62*
2.4.4 FILEコマンド群 VIII-63
• RF,RFDコマンド *VIII-64*
• VF,VFDコマンド *VIII-65*
• WF,WFDコマンド *VIII-66*
• CL,CSコマンド *VIII-67*
• OPTLDコマンド *VIII-68*
2.4.5 ROMコマンド群 VIII-69
• RPコマンド *VIII-70*
• VPコマンド *VIII-71*
• ROMコマンド *VIII-72*
2.4.6 CONTROLコマンド群 VIII-73
• I,TIMコマンド *VIII-74*
• OTFコマンド *VIII-75*
• Qコマンド *VIII-76*
2.4.7 HELPコマンド VIII-77
2.5 エラーメッセージ一覧 VIII-82
2.6 HEXファイルフォーマット VIII-83

1 機種による相違点

プログラムを作成するうえでは、S1C62 Familyの各機種によりメモリ容量が異なりますので、以下の点に注意する必要があります。
各機種の制限事項は"S5U1C62xxxD Manual"に記載されています。

■ ROM領域

機種によりROM容量が異なります。
ROMの最終アドレスを越えるICEコマンドの指定はエラーとなります。

■ RAM領域

機種によりRAM容量、使用領域が異なります。
RAMの最終アドレスを越えるICEコマンドの指定、および未使用領域の指定はエラーとなります。

■ 未定義コード

S1C62 Familyの中では機種による命令セットの違いはありません。ただし、SLP命令や、RAM容量の関係でインデックスレジスタのページ部(XP、YP)をアクセスする命令などが使用できない場合があります。
指定した場合はエラーとなります。

■ OPTLDコマンド

OPTLDコマンドはメロディHEXファイルなどをロードするコマンドで、使用できる機種が限られます。

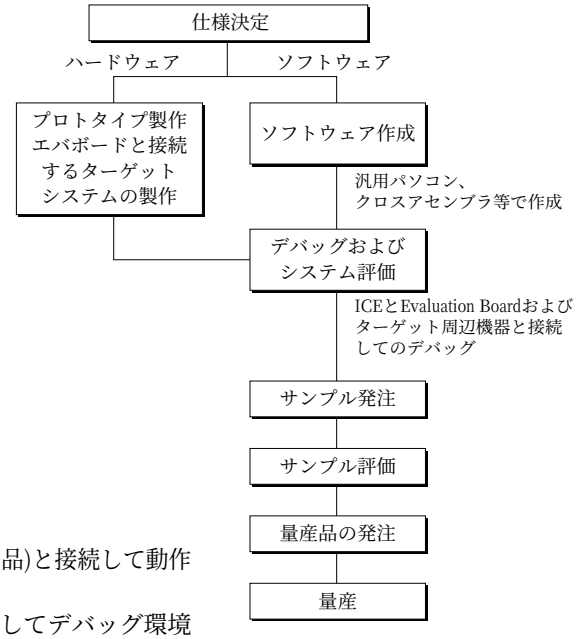
2 S5U1C62000H仕様

2.1 概要

ICE(S5U1C62000H)は、4-bit Single Chip Microcomputer S1C62 Familyのソフトウェア開発を効率よく行うためのマイクロコンピュータソフトウェア開発支援装置です。
ICEとS1C62 Family Evaluation Board(S5U1C62xxxE)の組み合わせにより、強力なハードソフト開発支援環境が提供されます。

S1C62 Familyの開発、量産の手順は右のフローで行われます。

ICEとEvaluation Boardにより、開発工程で最も時間のかかるデバッグおよびシステム評価工程を短縮することができます。



2.1.1 特長

ICEの特長を以下に示します。

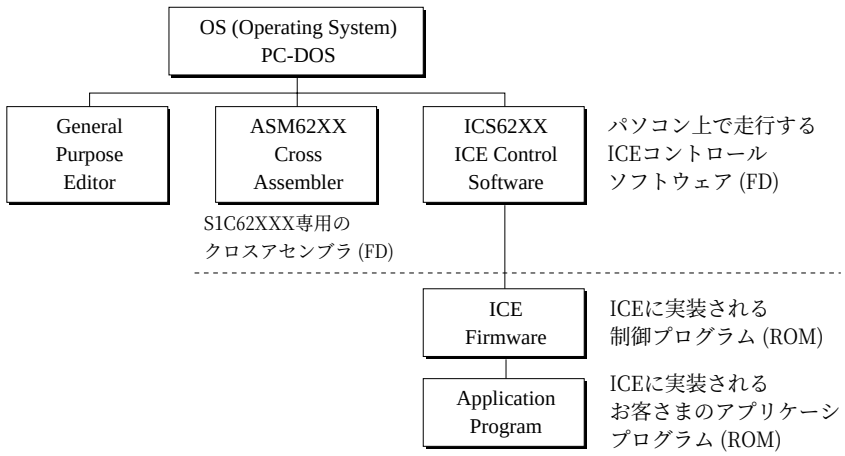
(1) 汎用パソコン(IBM社製 PC/XT、PC/ATおよび相当品)と接続して動作します。
お客さま手持ちのパソコンをホストシステムとしてデバッグ環境が構築されます。

(2) 高機能なエミュレーションコマンドを用意しています。
レジスタ値によるブレーク機能、オンザフライ情報表示、ヒストリ表示等高機能かつ豊富なコマンドを用意しています。

(3) ICE専用電源を装備しています。
本電源によりエバボードにV_{DD}を供給するため、お客さま側で電源ユニットの用意は不要です。

(4) ハードウェアの解析にも使用できます。
SYNC端子、HALT端子によりハードウェアデバッグもサポートしております。

2.1.2 ソフトウェア構成



2.1.3 機能一覧表

ICE(S5U1C62000H)の機能一覧表を表2.1.3.1に示します。

表2.1.3.1　ICE(S5U1C62000H)機能一覧表

項番	項　目	概　略　機　能	備　考
1	リアルタイム ブレーク	ターゲットプログラムを任意の条件で中断する (1) PCによるブレーク (2) RAMアドレス、データ、R/Wのブレーク (3) Evaluation Board CPUレジスタ値によるブレーク (4) (1)～(3)の複合(AND、OR)ブレーク (5) RESET、BREAKスイッチによる強制ブレーク (6) ホストのESCキー入力による強制ブレーク	
2	ヒストリ	エミュレーション中のEvaluation Board CPUの情報採取 (1) PC、命令コード、RAMのR/W、CPUレジスタ値の採取 (2) 約2048命令分のバス情報採取 (3) ブレーク条件ヒットまでの、またはヒット前後の情報採取 (4) 指定したプログラム領域のヒストリ情報採取 (5) ヒストリ情報のサーチ	
3	リアルタイム実行	ターゲットクロック4MHzまでのリアルタイム実行	
4	実時間測定	ランエミュレーション中の実時間(約425msecまで)または ステップ数のカウント	
5	ターゲットメモリの 参照、変更	(1) ICEに実装したターゲットプログラムメモリ、 データメモリの参照、変更、ダンプ (2) ターゲット上のメモリマップドI/Oの参照、変更 (3) Evaluation Board CPU内のレジスタの参照、変更	
6	トレース	ターゲットプログラムを1ステップずつ実行しレジスタ内容を表示	
7	アセンブル 逆アセンブル	ニーモニックキー入力のアセンブル(機械語に変換後プログラム メモリにストア) メモリ内容の逆アセンブル	
8	FDロード セーブ ベリファイ	(1) FDからプログラムメモリにプログラムロードまたはベリファイ (2) プログラムメモリからFDにプログラムセーブ (3) ICE途中結果をFDにロード、セーブ (4) FDからデータメモリにデータロード、セーブ、ベリファイ	
9	ROMリード ベリファイ	ICEのROMソケットからプログラムメモリにプログラムロード、 およびベリファイ	
10	実行監視	Gコマンド中の実行PCを表示、およびホールト状態の表示	
11	カバレッジ	カバレッジ情報の取得	
12	その他	(1) プリンタの起動、停止 (2) ICEコマンド表示 (3) Evaluation Board CPUのリセット (4) Evaluation Board CPUの状態をLED表示 (5) ブレークポイントでブレーク状態とせず、SYNCパルスを 出力しながらの実行 (6) 2764～27512のEPROM(ターゲット)サポート (7) ICEハードウェアのチェック	

2.1.4 機能別コマンド一覧表

ICEの機能別コマンド一覧表を表2.1.4.1に示します。

表2.1.4.1　機能別コマンド一覧表

項番	機　能	コマンド形式	動　作　概　要	ページ
1	アセンブル	#A,a ↵	命令ニーモニックをアセンブルしてa番地からストア	VIII-36
2	逆アセンブル	#L,a1,a2 ↵	a1番地からa2番地までの内容を逆アセンブル表示	VIII-20
3	ダンプ	#DP,a1,a2 ↵	プログラム領域a1～a2番地の内容表示	VIII-22
		#DD,a1,a2 ↵	データ領域a1～a2番地の内容表示	VIII-24
4	フィル	#FP,a1,a2,d ↵	プログラム領域a1～a2番地にデータdを設定	VIII-37
		#FD,a1,a2,d ↵	データ領域a1～a2番地にデータdを設定	VIII-38
5	ランモード設定	#G,a ↵	a番地からプログラムを実行	VIII-56
		#TIM ↵	実時間/ステップカウントの選択	VIII-74
		#OTF ↵	オンザフライ表示の選択	VIII-75
6	トレース	#T,a,n ↵	a番地からnステップを命令実行結果を表示しながら実行	VIII-58
		#U,a,n ↵	#T,a,nの最終ステップのみ結果表示	VIII-60
7	ブレーク	#BA,a ↵ #BAR,a ↵	プログラムアドレスaにブレークポイントを設定 ブレークポイントを解除	VIII-48
		#BD ↵ #BDR ↵	データRAMに関するブレーク条件の設定 #BDによるブレーク条件を解除	VIII-49
		#BR ↵ #BRR ↵	Evaluation Board CPU内レジスタに関するブレーク条件の設定 #BRによるブレーク条件を解除	VIII-50
		#BM ↵ #BMR ↵	プログラムアドレス・データRAM・レジスタ値による複合ブレーク条件の設定 #BMによるブレーク条件を解除	VIII-52
		#BRES ↵	すべてのブレーク条件を解除	VIII-55
		#BC ↵	ブレーク条件の表示	VIII-54
		#BE ↵	ブレークイネーブルモードの設定	VIII-61
		#BSYN ↵	ブレークディセーブルモードの設定	VIII-61
		#BT ↵	ブレークストップ/トレースモードの指定	VIII-61
		#BRKSEL,REM ↵	BA条件のクリア/リメインモードの指定	VIII-62
8	ムーブ	#MP,a1,a2,a3 ↵	プログラム領域a1～a2番地の内容をa3以降に転送	VIII-39
		#MD,a1,a2,a3 ↵	データ領域a1～a2番地の内容をa3以降に転送	VIII-40
9	データセット	#SP,a ↵	プログラム領域a番地からデータの書き込み	VIII-41
		#SD,a ↵	データ領域a番地からデータの書き込み	VIII-42
10	CPU内レジスタ変更	#DR ↵	Evaluation Board CPU内レジスタ内容の表示	VIII-26
		#SR ↵	Evaluation Board CPU内レジスタ内容の設定	VIII-43
		#I ↵	Evaluation Board CPUのリセット	VIII-74
		#DXY ↵	X, YとMX, MYの表示	VIII-34
		#SXY ↵	X, Yの表示とMX, MYのデータ設定	VIII-44

項番	機　能	コマンド形式	動　作　概　要	ページ
11	ヒストリ	#H,p1,p2 ↵	ポインタ1～ポインタ2のヒストリデータを表示	VIII-27
		#HB ↵	ヒストリデータをさかのぼって表示	VIII-30
		#HG ↵	ヒストリデータを21行分表示	VIII-30
		#HP ↵	ヒストリポインタを表示	VIII-33
		#HPS,a ↵	ヒストリポインタを設定	VIII-33
		#HC,S/C/E ↵	ブレークヒットの前(S)、前後(C)、後(E)のヒストリ情報採取を設定	VIII-44
		#HA,a1,a2 ↵	プログラム領域a1～a2番地のヒストリ情報採取を設定	VIII-45
		#HAR,a1,a2 ↵	プログラム領域a1～a2番地のヒストリ情報採取禁止を設定	VIII-45
		#HAD ↵	ヒストリ採取プログラム領域を表示	VIII-45
		#HS,a ↵	プログラムa番地を実行したヒストリ情報を検索して表示	VIII-32
		#HSW,a ↵ #HSR,a ↵	データ領域a番地をWRITEまたはREADしたヒストリ情報を検索して表示	VIII-32
12	ファイル	#RF,file ↵	プログラムファイルをメモリに転送	VIII-64
		#RFD,file ↵	データファイルをメモリに転送	VIII-64
		#VF,file ↵	プログラムファイルとメモリの内容を比較	VIII-65
		#VFD,file ↵	データファイルとメモリの内容を比較	VIII-65
		#WF,file ↵	メモリ内容をプログラムファイルにセーブ	VIII-66
		#WFD,file ↵	メモリ内容をデータファイルにセーブ	VIII-66
		#CL,file ↵	ファイルからICE設定値をロード	VIII-67
		#CS,file ↵	ICE設定値をファイルにセーブ	VIII-67
		#OPTLD,n,file ↵	HEXAデータをファイルからロード	VIII-68
13	カバレッジ	#CVD ↵	カバレッジ情報の表示	VIII-34
		#CVR ↵	カバレッジ情報のクリア	VIII-34
14	ROMアクセス	#RP ↵	ROM内容をプログラムメモリに転送	VIII-70
		#VP ↵	ROM内容とプログラムメモリ内容の比較	VIII-71
		#ROM ↵	ROMの種類の指定	VIII-72
15	ICEの終了	#Q ↵	ICEを終了しOSに制御を戻す	VIII-76
16	コマンド表示	#HELP ↵	ICEのコマンドを表示	VIII-78
17	自己診断	#CHK ↵	ICE自己診断結果の報告	VIII-33

↵ はリターンキーの入力を表します。

2.1.5 アルファベット順コマンド一覧表

ICEのアルファベット順コマンド一覧表を表2.1.5.1に示します。

表2.1.5.1　アルファベット順コマンド一覧表

項番	コマンド形式	動　作　概　要	ページ
1	#A,a ↵	命令ニーモニックをアセンブルしてa番地からストア	VIII-36
2	#BA,a ↵	プログラムアドレスaにブレークポイントを設定	VIII-48
3	#BAR,a ↵	ブレークポイントを解除	VIII-48
4	#BC ↵	ブレーク条件の表示	VIII-54
5	#BD ↵	データRAMに関するブレーク条件の設定	VIII-49
6	#BDR ↵	#BDによるブレーク条件を解除	VIII-49
7	#BE ↵	ブレークイネーブルモードの設定	VIII-61
8	#BM ↵	プログラムアドレス・データRAM・レジスタ値による複合ブレーク条件の設定	VIII-52
9	#BMR ↵	#BMによるブレーク条件を解除	VIII-52
10	#BR ↵	Evaluation Board CPU内レジスタに関するブレーク条件の設定	VIII-50
11	#BRR ↵	#BRによるブレーク条件を解除	VIII-50
12	#BRES ↵	すべてのブレーク条件を解除	VIII-55
13	#BRKSEL,REM ↵	BA条件のクリア/リメインモードの指定	VIII-62
14	#BSYN ↵	ブレークディセーブルモードの設定	VIII-61
15	#BT ↵	ブレークストップ/トレースモードの指定	VIII-61
16	#CHK ↵	ICE自己診断結果の報告	VIII-33
17	#CL,file ↵	ファイルからICE設定値をロード	VIII-67
18	#CS,file ↵	ICE設定値をファイルにセーブ	VIII-67
19	#CVD ↵	カバレッジ情報の表示	VIII-34
20	#CVR ↵	カバレッジ情報のクリア	VIII-34
21	#DD,a1,a2 ↵	データ領域a1～a2番地の内容表示	VIII-24
22	#DP,a1,a2 ↵	プログラム領域a1～a2番地の内容表示	VIII-22
23	#DR ↵	Evaluation Board CPU内レジスタ内容の表示	VIII-26
24	#DXY ↵	X, YとMX, MYの表示	VIII-34
25	#FD,a1,a2,d ↵	データ領域a1～a2番地にデータdを設定	VIII-38
26	#FP,a1,a2,d ↵	プログラム領域a1～a2番地にデータdを設定	VIII-37
27	#G,a ↵	a番地からプログラムを実行	VIII-56
28	#H,p1,p2 ↵	ポインタ1～ポインタ2のヒストリデータを表示	VIII-27
29	#HA,a1,a2 ↵	プログラム領域a1～a2番地のヒストリ情報採取を設定	VIII-45
30	#HAD ↵	ヒストリ採取プログラム領域を表示	VIII-45
31	#HAR,a1,a2 ↵	プログラム領域a1～a2番地のヒストリ情報採取禁止を設定	VIII-45
32	#HB ↵	ヒストリデータをさかのぼって表示	VIII-30
33	#HC,S/C/E ↵	ブレークヒットの前(S)、前後(C)、後(E)のヒストリ情報採取を設定	VIII-44
34	#HELP ↵	ICEのコマンドを表示	VIII-78
35	#HG ↵	ヒストリデータを21行分表示	VIII-30

項番	コマンド形式	動　作　概　要	ページ
36	#HP ↵	ヒストリポインタを表示	VIII-33
37	#HPS,a ↵	ヒストリポインタを設定	VIII-33
38	#HS,a ↵	プログラムa番地を実行したヒストリ情報を検索して表示	VIII-32
39	#HSR,a ↵	データ領域a番地をREADしたヒストリ情報を検索して表示	VIII-32
40	#HSW,a ↵	データ領域a番地をWRITEしたヒストリ情報を検索して表示	VIII-32
41	#I ↵	Evaluation Board CPUのリセット	VIII-74
42	#L,a1,a2 ↵	a1番地からa2番地までの内容を逆アセンブル表示	VIII-20
43	#MD,a1,a2,a3 ↵	データ領域a1～a2番地の内容をa3以降に転送	VIII-40
44	#MP,a1,a2,a3 ↵	プログラム領域a1～a2番地の内容をa3以降に転送	VIII-39
45	#OPTLD,n,file ↵	HEXAデータをファイルからロード	VIII-68
46	#OTF ↵	オンザフライ表示の選択	VIII-75
47	#Q ↵	ICEを終了しOSに制御を戻す	VIII-76
48	#RF,file ↵	プログラムファイルをメモリに転送	VIII-64
49	#RFD,file ↵	データファイルをメモリに転送	VIII-64
50	#ROM ↵	ROMの種類の指定	VIII-72
51	#RP ↵	ROM内容をプログラムメモリに転送	VIII-70
52	#SD,a ↵	データ領域a番地からデータの書き込み	VIII-42
53	#SP,a ↵	プログラム領域a番地からデータの書き込み	VIII-41
54	#SR ↵	Evaluation Board CPU内レジスタ内容の設定	VIII-43
55	#SXY ↵	X, Yの表示とMX, MYのデータ設定	VIII-44
56	#T,a,n ↵	a番地からnステップを命令実行結果を表示しながら実行	VIII-58
57	#TIM ↵	実時間/ステップカウントの選択	VIII-74
58	#U,a,n ↵	#T,a,nの最終ステップのみ結果表示	VIII-60
59	#VF,file ↵	プログラムファイルとメモリの内容を比較	VIII-65
60	#VFD,file ↵	データファイルとメモリの内容を比較	VIII-65
61	#VP ↵	ROM内容とプログラムメモリ内容の比較	VIII-71
62	#WF,file ↵	メモリ内容をプログラムファイルにセーブ	VIII-66
63	#WFD,file ↵	メモリ内容をデータファイルにセーブ	VIII-66

↵ はリターンキーの入力を表します。

2.2 システムの接続と起動

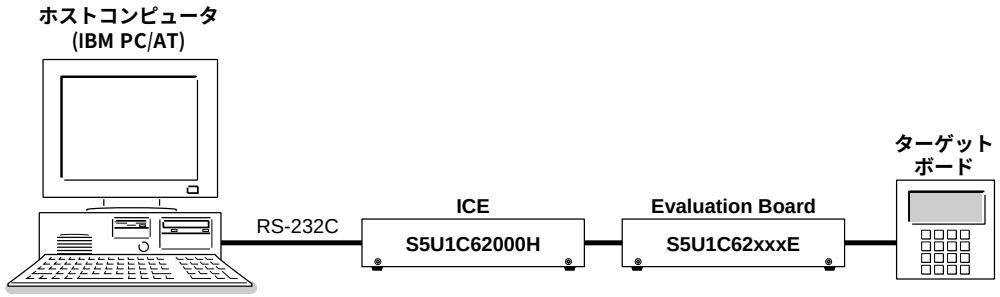


図2.2.1　システム接続図

ICEは図2.2.1のシステム接続図に示すようにS1C62 Family Evaluation Boardと汎用パソコンに接続して動作します。
システムの接続は下記の手順にしたがって行います。

(1) 電源オフを確認します。
汎用パソコン、ICEの電源がオフになっていることを確認します。
(S1C62 Family Evaluation BoardにはICEから電源を供給するため電源は内蔵されていません。)

(2) ケーブルを接続します。
"S5U1C62000H Manual"を参照してICEを汎用パソコンとEvaluation Boardに指定のケーブルで接続してください。

(3) 電源をオンにします。
汎用パソコン、ICEの電源をオンにします。順序はどちらが先でもかまいません。

2.2.1 ホストシステムの設定

ICEはRS-232Cインタフェースを通して汎用パソコン側から制御されます。また、ICS62XXシステムプログラムは約140KByteの容量があります。
したがって、ICEシステムを起動する前に汎用パソコンをICS62XXシステムプログラムが正常動作できる条件に設定する必要があります。

以下に例を示します。

■ プログラム(RAM)容量

ICS62XXシステムプログラム約140KByteが動作できるRAM容量を持ったパソコン構成としてください。

■ RS-232Cの設定

- **IBM社製PC/XT、PC/ATでICEを動作させる場合**

MODEコマンド …… PC-DOS起動後すぐに行ってください。

```
A>MODE COM1:4800,n,8,1,P↵          … 設定
COM1:4800,n,8,1,P                  … 設定内容を確認
A>
```

ICEのボーレートの切り換えは4800に設定してください。

2.2.2 ICS62XXの起動方法

■ オペレーティングシステムの起動

まず、汎用パソコンのオペレーティングシステム(以下、OSと略します)を立ち上げます。
ICS62XXが動作できるOSには下記のものがあります。

PC-DOS Version 2.1 以上

OSの立ち上げについては、それぞれのマニュアルを参照してください。
OSを立ち上げ後2.2.1項に示したホストシステムの設定を行ってください。

■ ICS62XXの起動

(1) ICS62XX ICE Control Software(CD-ROMで提供されます)をパソコンの指定ドライブにセットします。

(2) 次のようにキー入力します。

```
B>ICS62XX↵                    (アンダーラインはキー入力を示します)
…  初期画面を約1秒間表示します  …
* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *
#_
 ↑  カーソル位置
```

上記キー入力でICS62XXシステムプログラムがパソコン上にロードされ、ICS62XXシステムプログラムに制御が移ります。
正常に起動すると # を表示し、ICS62XXのコマンド待ちとなります。

■ ICS62XXの終了

```
#Q↵
B>
```

QコマンドでICS62XXプログラムを終了しOSに制御を戻します。

2.3 S5U1C62000Hの動作と機能

本項ではS5U1C62000Hの動作方式、機能の詳細、およびエミュレーション時の制限等について説明します。

2.3.1 動作概要

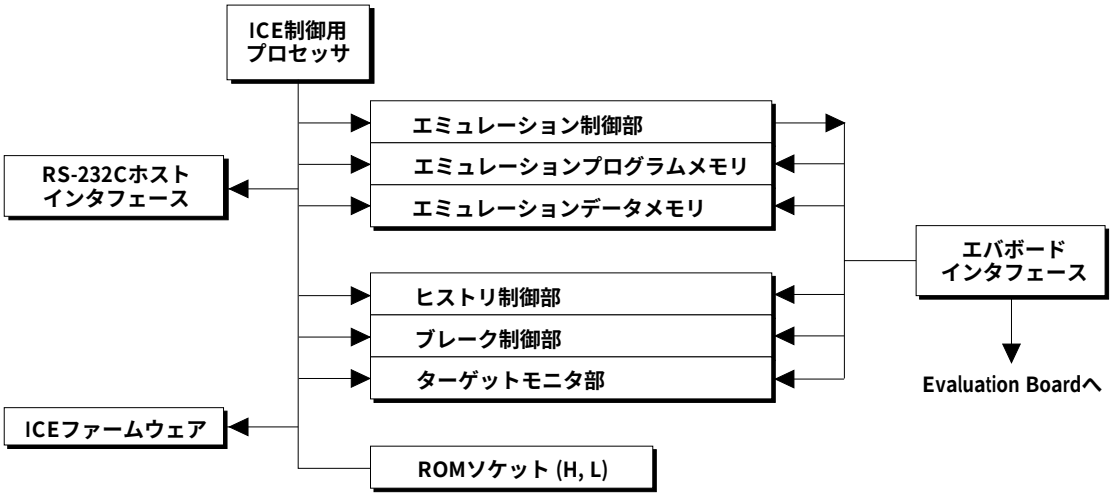


図2.3.1.1　ICE機能ブロック図

図2.3.1.1にICEの機能ブロック図を示します。
ICE筐体内にはICE制御用のプロセッサが内蔵されています。このプロセッサによりICEのコマンド処理が行われます。

Evaluation Board CPUを実行させたり停止させたりする機能をエミュレーションといい、この制御はエミュレーション制御部により行われます。
Evaluation Board CPUはラン(Gコマンドで指定)とシングルステップ(Tコマンドで指定)動作時以外は停止した状態となります。この状態を準備モードといい、ICE表示部のEMULATIONランプが消灯し、HALTランプが点灯します。ICEのAコマンド等の実行時も準備モードです。

エミュレーションプログラムメモリはEvaluation Board CPUが実行する命令が配置されます。
準備モードではICE制御用プロセッサによりROMソケットからのプログラムロード、ホストからのプログラム設定等が行われます。同様にエミュレーションデータメモリへはEvaluation Board CPUのデータRAMが割り付けられます。

ヒストリ制御部はEvaluation Board CPUの実行バスサイクルを記録する部分で、8192ワード×88ビットのメモリで構成されています。大容量のメモリを持つことによりリアルタイムにEvaluation Board CPU内のレジスタの値を記録します。ターゲットラン状態でヒストリ書き込みが行われ、準備モードでICE制御用プロセッサにより解析されます。

ブレーク制御部はEvaluation Board CPUのバス条件とブレークポイントを比較し、それらが一致したときに実行を停止させる制御を行います。CPUのレジスタ値によってもリアルタイムにブレークさせることができます。

ターゲットランの間ICE制御用プロセッサはターゲットモニタ部によりEvaluation Board CPUの実行PC値をモニタします。モニタ結果はオンザフライ情報として表示されます。

2.3.2 ブレークモードとブレーク機能

ブレークには以下のモードがあります。

- **ブレークイネーブルモード** … ブレーク機能を有効にします。
  ブレーク時の動作はブレークトレース/ストップのモード設定によって決まります。

- **ブレークディセーブルモード** … ブレーク機能を無効にします。
  ブレーク条件に一致した場合でもGコマンドを終了せずICEのSYNC端子にパルスを出力するモードです。
  本機能はこのパルスを基準としてターゲット上の回路のタイミングを測定したい場合などに、オシロスコープの同期信号として使用すると便利です。

- **ブレークトレースモード** … ブレーク条件に一致した場合ターゲットランを一時停止してEvaluation Board CPUレジスタ値と実行時間を表示した後すぐに再ランするモードです。
  リアルタイム性はありませんがプログラムのダイナミックな動作を見るときに有効です。

- **ブレークストップモード** … ブレーク条件に一致した場合にプログラムをブレークするモードです。

またブレークの種類には下記のものがあります。

- **リセットスイッチ** … ブレークモードの状態にかかわらずブレークします。
  ICEもリセットするためブレーク時のターゲットレジスタ値は表示されません。

- **ブレークスイッチ** … ブレークモードの状態にかかわらずブレークします。
  ブレーク時のEvaluation Board CPUのレジスタ値は正しく表示されます。

- **ESCキー** … ホストのESCキー入力によりブレークします。
  本機能もブレークモードの状態にかかわらずブレークします。ブレーク時のEvaluation Board CPUのレジスタ値は正しく表示されます。

- **ブレーク指定コマンド** … BA、BD、BR、BMコマンドにより指定された条件とEvaluation Board CPU条件が一致したときブレークします。
  ブレークイネーブルモード&ブレークストップモード時にブレークし、ブレークディセーブルモードではブレークしません。ブレークトレースモードではコマンド終了条件とはなりません。

ブレークモードとブレークの種類の機能表を表2.3.2.1に示します。

表2.3.2.1　ブレークモードとブレークの種類

項番	ブレークモード	ブレークさせる手段	説　明
1	ブレークイネーブル &ブレークストップ	・リセットスイッチ ・ブレークスイッチ ・ESCキー ・ブレーク指定コマンド	通常使用するモードです。 電源投入時も本モードになります。 本モード指定後のGOではEvaluation Board CPUはリアルタイムにランします。
2	ブレークイネーブル &ブレークトレース	・リセットスイッチ ・ブレークスイッチ ・ESCキー	ブレークトレース機能が実現されます。 BEコマンド、BTコマンドにより本モードになります。 ブレーク指定コマンドで設定した条件にEvaluation Board CPUが一致したときレジスタ情報を表示します。 本モード指定後のGOではEvaluation Board CPUはリアルタイムにランしません。 (レジスタ情報表示時にCPUは停止しています。)
3	ブレーク ディセーブル &ブレークストップ	・リセットスイッチ ・ブレークスイッチ ・ESCキー	SYNC出力機能が実現されます。 BSYNコマンドで本モードになります。 ブレーク指定コマンドで設定した条件にCPUが一致したときSYNC端子にパルスを出力します。 本モード指定後のGOではEvaluation Board CPUはリアルタイムにランします。
4	ブレーク ディセーブル &ブレークトレース	—	本指定は行えません。 ブレークディセーブルが指定されるとブレークストップモードに自動設定されます。 また、ブレークトレースが指定されるとブレークイネーブルモードに自動設定されます。

2.3.3 SYNC端子、HALT端子出力

SYNC端子出力

ブレーク条件と一致した命令サイクルの、次の命令のフェッチサイクル前半でLOWレベルのパルスが出力されます。

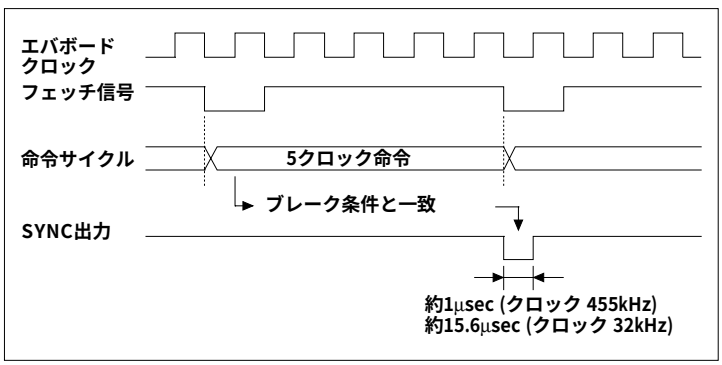


図2.3.3.1.a　SYNC端子出力

HALT端子出力

Evaluation Board CPUが停止しているとき(HALT、SLP命令実行)LOWレベルが出力されます。

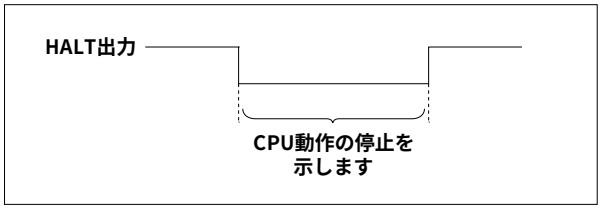


図2.3.3.1.b　HALT端子出力

2.3.4 ラン中の表示とブレーク時の表示

ラン中はICE制御用のプロセッサがEvaluation Board CPUの実行状態を監視します。
このとき、オンザフライ表示モード(OTFコマンドで指定します)が指定されているとEvaluation Board CPUの実行PCを約500msecごとにモニタ表示します。

```
#G↵
*PC=0120        ……  アンダーライン部にPCの値を次々に表示します。
*PC=HALT        ……  HALT状態になると改行し、HALTと表示します。
*PC=0200        ……  HALTが解除され動作が再開するとPC表示を再開します。
```

注: HALTはHALT命令、SLP命令の実行を示します。
プリンタを起動している場合、PCの値は次々に印字されます。
オンザフライ禁止モードにした場合PC表示は行われません。
ブレーク時はブレーク要因と、ブレーク後のPC(次に実行されるプログラムアドレス)とブレーク時のCPUレジスタ内容、および実行時間が表示されます。

```
#G↵
*PC=xxxx
*EMULATION END STATUS=BREAK HIT                    ……  (1)
*PC=0201 A=0 B=0 X=070 Y=071 F=IDZC SP=10          ……  (2)
*RUN TIME=425.097mS                                ……  (3)
```

(1) エミュレーション終了ステータスには BREAK HIT、ESC KEY、BREAKSWの3種類があります。
複数の条件がヒットした場合、BREAK SW>ESC KEY>BREAK HITの優先順位で高い優先順位のもののみ表示します。
リセットスイッチによってもブレークさせることはできますが、このときは
`* ICE6200 RESET SW TARGET *`
と表示しコマンド待ちになります。レジスタ表示、実行時間表示が行われませんので注意してください。

(2) 表示されるPCは次に実行する値を示しています。A以降のレジスタ値はブレーク時の値を示しています。
上記例では0200番地の命令終了後の値を示しています。

(3) RUN TIME表示には実行時間モードとステップ数モードの指定が可能です。
(#TIMコマンドで指定します。)
mSはミリセカンドを表します。ステップ数モードが指定された場合は、実行ステップ数を
`*RUN TIME=501 STEPS`
というように10進値で表示します。
実行時間またはステップ数カウンタがオーバフローしたときには
`*RUN TIME=TIMEOVER`
と表示します。
詳細は2.3.10項を参照してください。

2.3.5 ブレーク指定コマンド

ICEは豊富なブレーク指定機能を用意しています。

1 PCによるブレーク指定

BAコマンドで指定します。Evaluation Board CPUのPCと指定値が一致するとその命令を実行してブレークします。ただし、指定アドレスがPSET命令の場合、PSET命令とその次の命令を実行して停止します。(PSET命令が続く場合、PSET命令以外の命令が実行されたところで停止します。)
複数のPC(最大プログラムメモリサイズ分)に対しブレーク指定が可能です。

2 データRAMによるブレーク指定

BDコマンドで指定します。データRAMのアドレス・データ・R/Wのアンド条件でブレークします。
また、アドレス、データ、R/Wはそれぞれマスク指定が可能です。
10番地にデータFをライトしたときブレークさせるには、アドレス=10・データ=F・R/W=Wと指定します。データは何であってもよい場合にはアドレス=10・データ=マスク・R/W=Wと指定します。指定条件のメモリアクセスを行う命令実行後ブレークします。
本指定によるブレークポイントは1箇所のみ指定できます。

3 レジスタ値によるブレーク指定

BRコマンドで指定します。Evaluation Board CPUのレジスタ値がブレーク指定値と一致したとき、命令実行後ブレークします。
Aレジスタ・Bレジスタ・FI・FD・FZ・FC・Xレジスタ・Yレジスタのアンド条件でブレークします。また、各レジスタについてマスク指定が可能です。
A=5・X=70・Y=0Aのときにブレークさせるにはは A=5・X=70・Y=0A・その他のレジスタ=マスクと指定します。

```
例:     LD   A,5
        LD   X,70
        LD   Y,0A        …… この命令の実行後ブレークします。
```

本指定においてもリアルタイム性は保証されます。
ブレークポイントは1箇所のみ指定できます。

上記 1、2、3は独立に指定が可能です。
BA、BD、BRをすべて指定してランさせた場合、いずれかの条件が一致するとブレークします。

4 複合ブレーク指定

BMコマンドで指定します。複合ブレークは1、2、3のアンドが成立したときブレークします。

ブレーク成立=(PC一致)・(データRAMアドレス・データ・R/W一致)・(レジスタ値)一致

それぞれの要素はマスク指定が可能です。
ブレークポイントは1箇所のみ指定できます。
本指定が行われると1～3の指定は自動的に解除されます。
また、本指定のあと1～3の指定があるとBM条件は解除されます。

注: データRAM条件がブレークの要素としてあるため、データRAMアクセスを行う命令以外ではブレークしません。

2.3.6 ターゲット割り込みとブレーク

ブレークする瞬間にターゲット割り込みが発生した場合、ターゲット割り込みを優先させ、割り込み処理のスタック操作を行った後ブレークします。次にランを実行すると割り込み処理ルーチンの先頭から実行します。
ブレーク時に表示されるPCは割り込み処理先頭アドレスを示します。
BRコマンドでFI=1でブレーク指定されている場合、ブレークと割り込みが同時に発生するとFI=1でブレークしますが、割り込み処理を行ったためブレーク後のレジスタ表示は

```
*PC=0000  A=....  F=.DZC  X=000  Y=010
                     |
                  FIリセット
```

のようにFIフラグがリセットされた状態となります。

2.3.7 ヒストリ機能

ランエミュレーション中Evaluation Board CPUの情報(PC、命令コード、データRAMアドレスとデータ内容、CPU内レジスタ値)はヒストリメモリと呼ばれる領域にCPUのバスサイクルごとに取り込まれます。ヒストリメモリは8192サイクル分の容量があり、命令単位で5クロック命令のみの場合は最大2730命令、12クロック命令のみの場合は最大1365命令までEvaluation Board CPUが実行した最新の命令をストアすることができます。

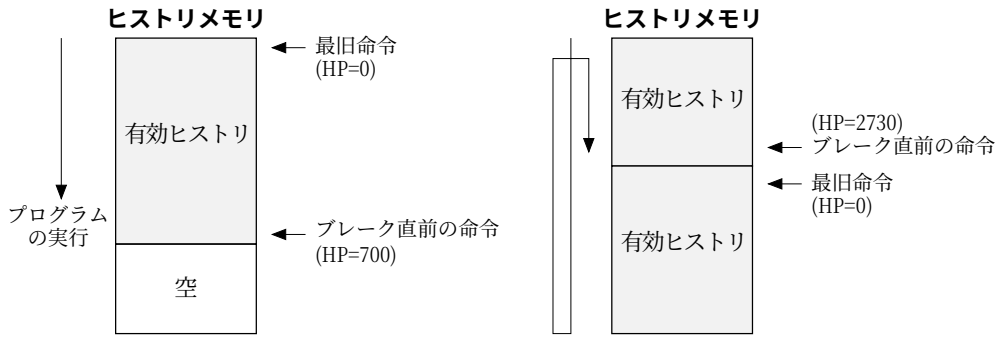
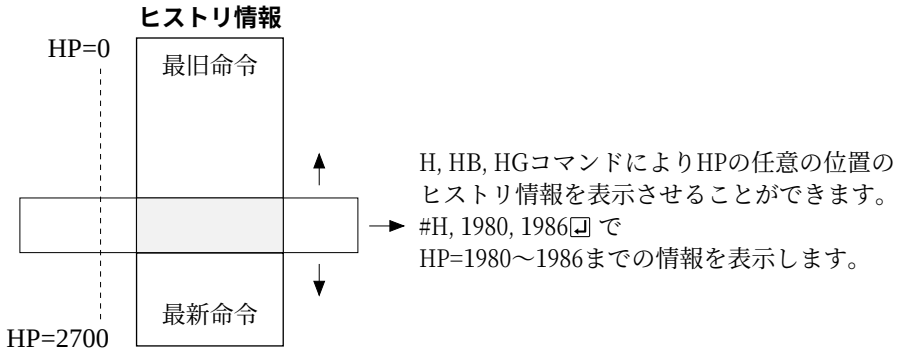


図2.3.7.1　ヒストリ機能図

図2.3.7.1はヒストリ機能図を示します。ヒストリメモリは書き込みで満杯になると古い情報を消して新しい情報をオーバライトしていきます。
HPはヒストリポインタと呼ばれ、0の位置が最も古い命令、ブレーク時の値の位置が最も新しい命令を示しています。HPの最大値は5クロック命令のみの実行時で約2730となります。




```
#H, 1980, 1986↵
LOC   PC  IR OP    OPR. A B   X   Y IDZC MEMORY OPERATION    OTHER
1980 0200 FC1 PUSH B     0 0 03F 03F 1111 W010=0
1981 0201 423 CALL 23    0 0 03F 03F 1111 W00F=8 W00E=0 W00D=2  ......(1)
1982 0223 FDF RET        0 0 03F 03F 1111 R00D=2 R00E=0 R00F=8
1983 0202 FD1 PDP  B     0 0 03F 03F 1111 R010=0
*1984                                     W010=8 W00F=0 W00E=2 INT1
1985                                                          INT2
1986 00FE FFF NOP7      0 0 0FF 0FF 0111                      
```

(a) (b) (c) (d) (e) (f) (g) (h)

(a) ヒストリポインタを示します。

(b) 実行された命令アドレスを示します。

(c) 命令コードを示します。

(d) 命令のニーモニックを表示します。

(e) 命令終了時のレジスタの値を表示します。

(f) 各フラグがセットされているときに"1"、リセットされているときに"0"を表示します。

(g) 命令実行中にデータメモリの書き込みまたは読み出しが行われた場合に、そのR/Wアドレス、データの順に表示します。(1)は0F番地に"8"、0E番地に"0"、0D番地に"2"が書き込まれたことを示します。

(h) 割り込み処理があった場合"INT1"(スタック処理)、"INT2"(ベクタ処理)と表示します。
INT1のMEMORY OPERATIONはスタックサイクルを示しています。

注: * 割り込み処理が行われた場合、HPは2個更新されます。それ以外はHPは命令単位に更新されます。

2.3.8 ブレーク遅延機能

前項で説明したヒストリ機能により、ブレークに至るまでのプログラムの動きをさかのぼって参照することができます。ICEでは、この機能を拡張してブレーク条件をヒットした後の、あるいはブレーク条件をヒットした前後のヒストリ情報を取得し参照することができます。この機能を実現するためにブレーク条件をヒットした後すぐにプログラムを停止せずに、所定量のヒストリデータを採取するまでプログラムを実行した後ブレークするようにしています。この指定は#HCコマンドによって行います。

注: ブレークイネーブル&ブレークストップモード(2.3.2項参照)でブレーク遅延を指定した場合、指定したブレーク条件のところではブレークしなくなることに注意してください。

2.3.9 カバレッジ機能

ICEは、プログラムの実行時にアクセスしたプログラムの番地情報を採取し表示することができます。プログラムを長い間実行させた後にカバレッジ情報を参照して、プログラムを実行した部分と実行していない部分を知ることにより、障害解析やデバッグが済んだ部分と済んでいない部分の確認などが行えます。このカバレッジ機能は#CVD、#CVRコマンドによって指定します。

2.3.10 実行時間の測定

ICS62XXにはターゲットプログラムがランしてからブレークするまでの時間、またはステップ数をカウントする機能があります。

測定範囲は次のようになります。

時間測定モード

6.5µsec ～ 6.5 × 65535µsec (=425.977msec)

測定誤差…±6.5µsec

表示はミリ秒(msec)単位で行われます。

ステップカウントモード

1ステップ ～ 65535ステップ

測定誤差… 0ステップ (ただし割り込み処理があった場合、これを1ステップとみなします。)

測定範囲を越えてブレークした場合、
"* RUN TIME=TIMEOVER"と表示します。

2.3.11 自己診断機能

ICEはパワーオン時にセルフチェックを行います。
ホストからチェックコマンド(#CHK↵)が入力されるとセルフテストの結果をホストに転送します。

```
#CHK↵
#                                   … エラーがないときコマンド待ちになります。
```

また、ICS62XXシステムプログラムロード時にチェックコマンドは自動的に実行されます。

```
B>ICS62XX↵
…  初期画面の表示  …
* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *  … チェックコマンドが自動的に実行され異常が
#                          なければ左記メッセージを表示 します。
```

上記表示によりICEとホストの接続、ICEの基本動作が正常であることがわかります。

ケーブルが接続されていなかったり、ICEの電源がオフの場合

```
B>ICS62XX↵
* COMMUNICATION ERROR OR ICE NOT READY *
```

と表示し、ICE側の動作を待ちます。ここでICEに電源が投入されると

```
* ICE POWER ON RESET *
* DIAGNOSTIC TEST OK *
#
```

と自動的にテストを行います。
チェックコマンドでエラーが表示されたときはハードの故障と考えられますのでご連絡ください。

2.3.12 プリンタの起動

プリンタの起動はOSの制御下で行われます。

ICS62XXシステム走行中でもCTRL+Pキーによりプリンタの起動、停止が行えます。

```
#BA,100↵
#CTRL+P T↵          … CTRL+P以降の画面文字が印字されます。
PC=300 IR=FFF ..... SP=010
#
  :
#CTRL+P             … プリンタ印字を終了させます。
```

2.3.13 エミュレーション中の制限事項

ICEとEvaluation Boardを接続しエミュレーションを行うとき、2.3.1項に示したように通常はEvaluation Board CPUは停止した状態にあります(準備モード)。
準備モードにおいてはEvaluation BoardはCore CPUも周辺も停止しており、不当な動作は起きません。準備モードが解除され、ターゲットプログラムを実行するまでにEvaluation Board CPUはコマンドプログラムと呼ぶICE制御のための命令を実行します。エミュレーションが終了し準備モードに移るときもコマンドプログラムを実行します。

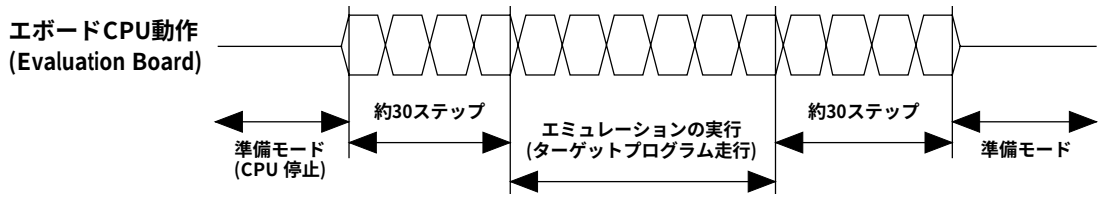


図2.3.13.1　Evaluation BoardのCPU動作

タイマ、ウォッチドッグタイマ等を内蔵する機種ではICEのコマンドプログラムが動きだ し、タイマ等がイネーブルになっているとその更新を始めます。また、ICEはコマンドプログラムの中でエミュレーションに移る直前にウォッチドッグをクリアします。
したがって、ICEを使用する場合下記の事項に注意してください。

(1) トレースコマンド(T、U)を長く実行させる場合

Evaluation Board上のタイマ値はコマンドプログラムの実行時に更新される可能性があります。

(2) ランを停止し再ランする場合

ウォッチドッグタイマはエミュレーション前後にICEがクリアするためウォッチドッグタイマの連続性はありません。
ランが十分長い場合はターゲットプログラム中でリアルタイムに働きます。

コマンドプログラムはエミュレーション前後共に約30ステップ走ります。
32kHzクロックで動く場合エミュレーション前後で6msec+6msec=12msecかかります。
455kHzクロックで動く場合エミュレーション前後で400μsec+400μsec=800μsecかかります。

なお、ダンプデータコマンド(#DD)でI/Oエリアの割り込みフラグを読み出してもフラグがクリアされることはありません。

2.4 コマンドの詳細

本項ではICEコマンドの機能詳細と使用方法について説明します。
コマンドは以下の6種類の属性に分けられています。

- **DISPLAYコマンド** … プログラムメモリ、データメモリ内容の表示や、ヒストリ情報の表示を行うコマンド群です。
- **SETコマンド** … メモリ内容(プログラムメモリ、データメモリ)の変更を行うコマンド群です。
- **BREAK & GOコマンド** … ブレーク条件の設定やエミュレーションを行うコマンド群です。
- **FILEコマンド** … HOST上のファイルとICE間の転送を制御するコマンド群です。
- **ROMコマンド** … ICEに実装するEvaluation Board CPU用ROM(LOW、HIGH)とプログラムメモリ間の転送を制御するコマンド群です。
- **CONTROLコマンド** … ICEの動作モードの指定コマンド群です。
(ターゲットのイニシャライズも含みます。)

使用例に用いられているプログラムは、S1C6S3N7/6S3B7/6S3L7対応となっています。
したがって、機種により実際のエラー出力等が異なりますので注意してください。

2.4.1項からのコマンド説明でコマンドの入力形式については下記の内容にしたがいます。

- コマンド待ち時は # が表示されます。
- コマンド入力は大文字、小文字とも許されます。
- コマンドで指定子 <　> がある場合","(カンマ)またはスペースで区切って入力します。
- コマンドフォーマットの項で(ガイダンス付)とあるコマンドはそのコマンドの中で対話形式でキー入力することを示します。
このとき、対話形式でキー入力する部分をアンダーラインで説明しています。
- コマンドフォーマットの項で(トグル)とあるコマンドはコマンド入力ごとに指定が反転することを示しています。
- 注意事項の欄はそのコマンドを使用するにあたっての注意点や留意点を示しています。

2.4.1 DISPLAYコマンド群

L	DISASSEMBLE LIST	VIII-20
DP	DUMP PROGRAM	VIII-22
DD	DUMP DATA RAM	VIII-24
DR	DISPLAY CPU REGISTER	VIII-26
H	HISTORY DATA DISPLAY	VIII-27
HB	HISTORY DATA DISPLAY BACKWARD	VIII-30
HG	HISTORY DATA DISPLAY FORWARD	VIII-30
HS	HISTORY SEARCH PC	VIII-32
HSR	HISTORY SEARCH MEMORY READ	VIII-32
HSW	HISTORY SEARCH MEMORY WRITE	VIII-32
HP	HISTORY POINTER DISPLAY	VIII-33
HPS	HISTORY POINTER SET	VIII-33
CHK	CHECK ICE HARDWARE	VIII-33
DXY	DISPLAY X, Y REGISTER & MX, MY CONTENT	VIII-34
CVD	DISPLAY COVERAGE	VIII-34
CVR	RESET COVERAGE	VIII-34

L *DISASSEMBLE LIST*

Format

#L,<アドレス1>,<アドレス2>↵
#L,<アドレス1>↵
#L↵

Function

プログラム領域(エミュレーションプログラムメモリ)の<アドレス1>番地から<アドレス2>番地までを逆アセンブル表示します。

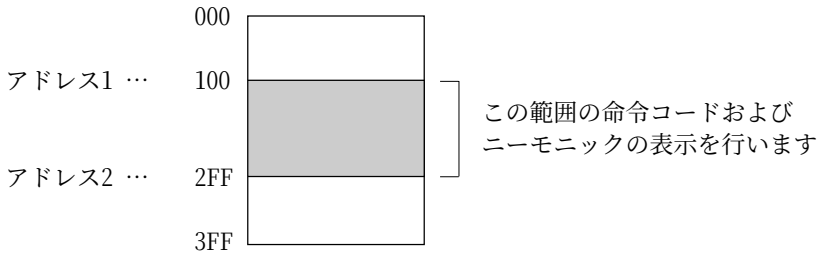
(1) <アドレス2>を省略した場合1画面分(22行)の逆アセンブル表示を行います。

(2) <アドレス1><アドレス2>とも省略した場合、前回の最終アドレス+1番地から1画面の逆アセンブル表示を行います。
電源投入後L↵のみの入力の場合には0番地から表示します。

(3) 1画面分以上の逆アセンブル表示を行う場合は、22行ごとに1行のスペースと約1秒のポーズをとりながら表示していきます。

(4) ESC入力によりコマンドを中断することができます。

プログラム領域 (S1C6S3N7/6S3B7/6S3L7の場合)



DISASSEMBLE LIST L

Format

#L,<アドレス1>,<アドレス2>↵
#L,<アドレス1>↵
#L↵

Examples

```
#L,100,1FF↵
 0100 FDF RET
 0101 2FF JP C,FF
   :   :   :
 01FF FFF NOP7
```
..... アドレス100～1FF番地のプログラム内容の逆アセンブル表示を行います。

```
#L,200↵
 0200 E00 LD A,0
 0201 E6F LDPX MX,F
   :   :   :
 0215 FFF NOP7
```
..... アドレス200番地から22行分の表示を行います。

```
#L↵
 0216 FDF RET
 0217 E05 LD A,5
   :   :   :
 022B FFB NOP
```
..... 前回の停止アドレス+1番地から22行分の表示を行います。

```
#L,100,FFF↵
 0100 FDF RET
   :   :   :
 0201 E6F LDPX MX,F
```
..... ESC入力により中断します。

```
#L,100,50↵
 * COMMAND ERROR *
```
..... アドレス1>アドレス2のためエラーとなります。

```
#L,100,100↵
 0100 FDF RET
```
..... アドレス100番地の内容を逆アセンブルします。正常実行されます。

```
#L,3FC↵
 03FC E00 LD A,0
   :
 03FF 20F JP C,F
```
..... プログラムエリアの最終アドレス(S1C6S3N7/6S3B7/6S3L7の場合3FF)を越えた場合、コマンドは終了します。

```
#
```

DP *DUMP PROGRAM*

Format

#DP,<アドレス1>,<アドレス2>↵
#DP,<アドレス1>↵
#DP↵

Function

プログラム領域(エミュレーションプログラムメモリ)の<アドレス1>番地から<アドレス2>番地までの内容を16進表示します。

(1) <アドレス2>を省略した場合、<アドレス1>番地から1画面分(21行分、21×8=168アドレス分)表示します。

(2) <アドレス1>、<アドレス2>共省略した場合、前回の最終アドレス+1番地から1画面分表示します。電源投入後、DP↵のみの入力の場合には0番地から表示します。

(3) 1画面分以上の表示を行う場合は21行ごとに1行のスペースと約1秒のポーズをとりながら表示します。

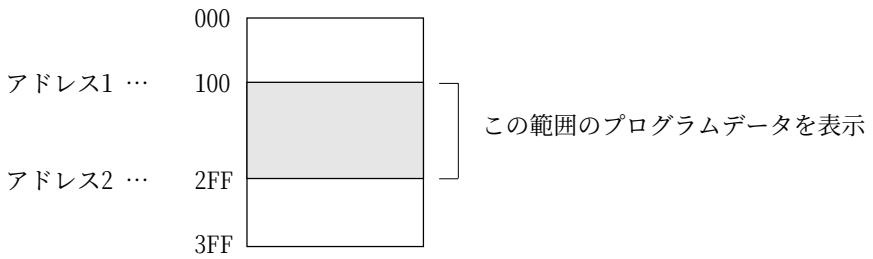
(4) 16進データと共にASCIIコードが表示されます。ASCIIコードはデータ内容がRETD、LBPX命令に限り、そのオペランド部をASCIIコードに変換して表示します。

例: データ内容142 … ASCII表示 B
(命令: RETD 42)

(5) 表示がプログラム領域の最終アドレスを越えた場合はそこでコマンドを終了します。

(6) ESC入力によりコマンドを中断することができます。

プログラム領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Format

#DP,<アドレス1>,<アドレス2>↵

#DP,<アドレス1>↵

#DP↵

Examples

```
#DP,104,121↵                    ..... 指定範囲を表示します。
 ADDR   0    1    2    3    4    5    6    7   ASCII
 0100                      FFF  FFB  930  142      ..0B
 0108  FFF  FFF  FFF  FFF  FFB  931  142  944  .....1BD
   :    :    :    :    :    :    :    :    :
 0118  FFF  FFF  FFF  FFF  FFB  FFB  FFB  FFB  ........
 0120  131  145                                 1E

#DP↵                            ..... 21行分表示します。
 ADDR   0    1    2    3    4    5    6    7   ASCII
 0120           131  132  145  FFF  FFB  FFB    12E...
   :    :    :    :    :    :    :    :    :
   :    :    :    :    :    :    :    :    :
                          21行分表示

#DP,0,FFF↵
 ADDR   0    1    2    3    4    5    6    7   ASCII
 0000  FFF  FFF  FFF  FFF  FFF  FFF  FFF  FFF  ........
   :    :    :    :    :    :    :    :    :
   :    :    :    :    :    :    :    :    :
                          ..... [ESC]入力によりコマンドを中断します。

#DP,100,50↵                     ..... アドレス1>アドレス2のためエラーとなります。
 * COMMAND ERROR *

#DP,400,FFF↵                    ..... プログラム領域の最大値(S1C6S3N7/6S3B7/6S3L7の場
                                      合3FF)を越えた指定のためエラーとなります。
 * COMMAND ERROR *

#
```

DD *DUMP DATA RAM*

Format

#DD,<アドレス1>,<アドレス2>↵
#DD,<アドレス1>↵
#DD↵

Function

データRAM領域の<アドレス1>番地から<アドレス2>番地までを16進表示します。

(1) <アドレス2>が省略された場合は、1画面分(21行分またはそれ以前にRAMの最終アドレスがある場合には最終アドレスまで)を表示します。

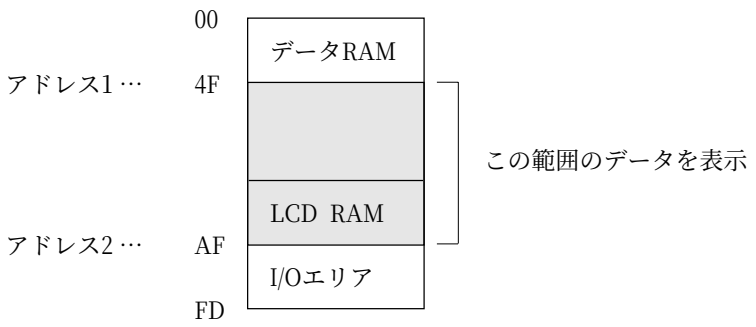
(2) <アドレス1>、<アドレス2>が省略された場合には前回の最終アドレス+1番地から1画面分を表示します。電源投入後のDD↵は0番地から表示します。

(3) ライトオンリーのI/Oエリアの内容は読み出しません。

(4) R/Wが混在するI/O番地は内容を読み出し、データおよび"!"のマークを表示します。

(5) ESC入力によりコマンドを中断することができます。

データRAM領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Format

#DD,<アドレス1>,<アドレス2>↵

#DD,<アドレス1>↵

#DD↵

Examples

```
#DD,80,BE↵
 ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
 0080 5 2 3 4 A B B C D 0 F F F F F F
 0090 - - - - - - - - - - - - - - - -
 00A0 - - - - - - - - - - - - - - - -  ..... ライトオンリーエリアはハイフン
 00B0 5 A 3 F 0 5 6 F 4 4 4 0 5 A A        表示となります。

#DD,100,FFF↵            ..... RAMアドレスが最大値(S1C6S3N7/6S3B7/6S3L7の場合7E)
                              を越えるためエラーとなります。
 * COMMAND ERROR *

#DD,0↵
 ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
 0000 F F F F F 0 0 0 0 0 0 1 1 1 2 3
   :                                 :
 00AF 5 A 3 F 0 5 6 F 4 4 4 0 5 A A
                        ..... 21行またはRAMの最終アドレスまで表示します。

#DD↵                    ..... 最終番地を超えたため再度0番地から表示します
                              (上記と同じ)。

#DD,50,40↵
 * COMMAND ERROR *      ..... アドレス1>アドレス2のためエラーとなります。

#DD,0,7E↵
 ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
 0000 F F F F F 0 0 0 0 0 0 1 1 1 2 3
   :
                        ..... [ESC]入力によりコマンドを終了します。

#DD,E40,F1F↵
 ADDR 0 1 2 3 4 5 6 7 8 9 A B C D E F
 0E40 F 0 1 5 7 4 A 0 0 0 E F 3 2 0 1
                                       ..... 未使用領域が1行全部の場合は、そ
 0E80 0 0 3 2 7 6 C 1 1 2 0 0 6 5 4 9        の行をスキップして表示します。
 0E90 1 5 7 6 C F 3 2 0 1 0 1 E A C 0        (S1C6S460の場合)
 0EA0 0 0 0 1 4 0 5 0 0 0 3 0 0 1 5 2
 0EBC 4 3 2 7 6 B A 0 1 5 D 3 2 7 4 3
 0EC0 5 5 4 1 0 2 3 6 0 0 0 1 5 6 7 F
 0F00 ! ! ! ! ! ! / / / / / / / / / /  ..... 表示行内のアドレスが未使用の場
 0F10 F 0 1 0 F F / / / / / / / / / /       合、そのアドレスはスラッシュ表
                                            示となります。 (S1C6S460の場合)
#
```

Note

ライトオンリー指定のI/Oのアドレスは読み出し動作を行いません。

DR *DISPLAY CPU REGISTER*

Format

#DR↵

Function

Evaluation Board CPUの現在のレジスタ値を表示します。

(1) PC … 次にエミュレーションを開始するアドレスを示します。

(2) A、B、X、Y、F、SP … 現在の値(ブレーク後ならブレーク時の値)を表示します。

(3) IR、ニーモニック … PCの示すプログラム領域の命令コードとニーモニックを表示します。

Example

```
#DR↵
 * PC=0100 IR=FFF NOP7 A=0 B=0 X=06F Y=03A F=IDZC SP=10
                                              |
#                                  F(フラグ)はセット時にフラグの頭文字を表示、
                                   リセット時は"."を表示します。
```

HISTORY DATA DISPLAY H

Format

#H,<ポインタ1>,<ポインタ2>↵
#H,<ポインタ1>↵

Function

ヒストリ情報を表示します。

(1) <ポインタ1>から<ポインタ2>までのヒストリ情報を表示します。

(2) <ポインタ2>省略時は<ポインタ1>から21行分のヒストリ情報を表示します。

(3) <ポインタ1>、<ポインタ2>で示される数は0～9999の10進数です。

(4) 命令ごとに下記内容を表示します。

- LOC … ヒストリポインタ 10進表示
- PC … プログラムカウンタ 16進表示 ブレークヒット時"[PC]"を表示
- IR … 命令コード 16進表示
- OP … 命令ニーモニック
- OPR … 命令オペランド
- A、B、X、Y … A、B、(XP、XH、XL)、(YP、YH、YL)レジスタの内容
- IDZC … フラグの内容(セット時"1"、クリア時"0")ビット対応の2進表示
- その他の情報 … 命令実行中メモリライト、リードサイクルがあった場合、そのサイクルとR/Wを表示。また、データ割り込みがあった場合、INT1(スタック情報)とINT2を表示。

(5) ヒストリメモリは8192バスサイクル分の容量を有します。一方S1C62 Familyの命令は5、7、12クロック命令があり、5クロック命令が3バスサイクル、7クロック命令が4バスサイクル、12クロック命令が6バスサイクルで構成されています。したがって、ヒストリポインタの最終値は実行された命令によって値が変わります。5クロック命令のみ実行時の最終値の最大は、約2700となります。12クロック命令のみ実行時は約1300となります。ヒストリメモリの最終に到るまでにブレークが成立するとヒストリポインタの最終値はさらに小さくなります。

(6) ヒストリメモリはブレークするまでの最新の情報を採取します。ヒストリ情報が(5)で説明したヒストリメモリの容量を越えた場合、最も古い情報から消去され、最新の情報がオーバーライトされます。

(7) ヒストリポインタの先頭は0です。
<ポインタ2>で最終値を越える指定があった場合最終値までを表示します。

(8) ヒストリデータがない場合(GOコマンド実行前、GOコマンドの実行後Tコマンド実行時、HARコマンド実行時)`* NO HISTORY DATA *`を表示します。

(9) ブレーク直前のヒストリを表示させる場合はHBコマンドを使用すると便利です。

H *HISTORY DATA DISPLAY*

Format

#H,<ポインタ1>,<ポインタ2>↵
#H,<ポインタ1>↵

Examples

#H,200,205↵ ..... 指定範囲を表示します。

```
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION   OTHER
 0200  0128 FDO POP  A     F 0 020 021 0011 R01F=0
 0201  0129 F70 DEC  M0    0 0 020 021 0010 R000=1 W000=0
 0202  012A 722 JP   NZ,22 0 0 020 021 0010
 0203  012B F71 DEC  M1    0 0 020 021 0000 R001=2 W001=1
 0204  012C 721 JP   NZ,21 0 0 020 021 0000
 0205  0121 F80 LD   M0,A  0 0 020 021 0000 W000=0
```

#H,300↵ ..... 21行分表示します。

```
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION   OTHER
 0300  000F C1F ADD  B,OF  F 4 02D 031 0001
 0301  0010 70E JP   NZ,OE F 3 02D 031 0001
 0302  000E EE8 LDPX MX,A  F 3 02D 031 0001 W02D=F
   :     :   :     :              :
 0319  0124 E10 LD   B,00  F 0 030 031 0001
 0320  0125 BD0 LD   X,D0  F 0 010 031 0001
```

#H,0,100↵

```
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION   OTHER
 0000  0000 E1C LD   A,B   5 4 000 024 0000
 0001  0001 E16 LD   B,06  4 4 000 024 0000
 0002  0002 822 LD   Y,22  4 6 000 022 0000
 0003  0003 EF0 INC  Y     4 6 000 022 0000
 0004  0004 EF3 LDPY A,MY  4 6 000 023 0000 R023=0
 0005  0005 90A LBPX MX,0A 0 6 001 024 0000 W000=A W001=0
 0006  0006 C05 ADD  A,05  0 6 002 024 0000
 0007  0007 D52 SBC  B,02  5 6 002 024 0000
 0008* 0008 17F RETD 7F   5 4 003 024 0000 R01A=C R01B=9 R01C=1 W002=F W003=7
```

* ヒストリメモリの最終を越えた場合コマンドは終了します。

#H,310,3000↵

```
  LDC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION   OTHER
 0310  0010 70E JP   NZ,0E F 0 020 021 0011
 0311  0011 8F1 LD   Y,21  F 0 020 021 0011
 0312  0012 E38 LD   MY,08 F 0 020 021 0011 W021=8
   :     :   :   :        : :  :   :    :
 2430  0172 E32 LD   MY,02 7 6 024 026 0000 W026=2
 2431  0173 F48 EI         7 6 024 026 0000
 2432  0174 FF8 HALT       7 6 024 026 1000
 2433                                      W01F=1 W01E=7 W01D=5 INT1
 2434                                                           INT2
 2435* 0108 0E6 JP   E6    7 6 024 026 0000
```

..... 割り込みがあった場合、INT1、INT2を表示します。

HISTORY DATA DISPLAY H

Format

#H,<ポインタ1>,<ポインタ2>↵

#H,<ポインタ1>↵

Examples

```
#H,0,500↵
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION   OTHER
 0000  0010 70E JP   NZ,0E F B 015 021 0001
 0001  000E EE8 LDPX MX,A  F B 015 021 0001 W015=F
 0002  000F C1F ADD  B,0F  F B 016 021 0001
 0003  0010 70E JP   NZ,0E F A 016 021 0001
 0004  000E EE8 LDPX MX,A  F A 016 021 0001 W016=F
 0005  000F C1F ADD  B,0F  F A 017 021 0001
 0006  0010 70E JP   NZ,0E F 9 017 021 0001
 0007  000E EE8 LDPX MX,A  F 9 017 021 0001 W017=F
 0008  000F C1F ADD  B,0F  F 9 018 021 0001
 0009  0010 70E JP   NZ,0E F 8 018 021 0001
 0010  000E EE8 LDPX MX,A  F 8 018 021 0001 W018=F
                     ..... [ESC]入力によりコマンドを終了します。
#
```

Note

ヒストリ情報のレジスタの値は、命令実行の次の行で変化します。
ただし、"LD X,x"および"LD Y,y"にかぎり命令実行の行で変化します。

HB, HG *HISTORY DATA DISPLAY BACKWARD/FORWARD*

Format

#HB↵

#HG↵

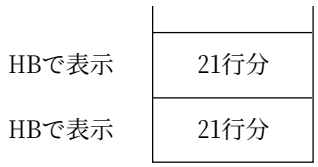
Function

ヒストリポインタ前後のヒストリ情報を表示します。

(1) HB … カレントヒストリポインタからさかのぼって21命令分を表示します。
表示後カレントヒストリポインタを-21します。
(ヒストリデータの最終値付近の表示に有効です。)

(2) HG … カレントヒストリポインタから21命令分表示します。
表示後カレントヒストリポインタを+21します。
(ヒストリデータを画面単位に古い情報から表示させるときに有効です。)

(3) GOコマンド終了時カレントヒストリポインタは、最終ヒストリポインタを指しています。



← カレントヒストリポインタ=最終ヒストリポインタ-42
(HB2回実行時)
← カレントヒストリポインタ=最終ヒストリポインタ-21
(HB1回実行時)
← カレントヒストリポインタ=最終ヒストリポインタ
(GO終了直後)

Examples

```
#BA,108↵

#G,R↵
 *PC=
 *PC=HALT
 *EMULATION END STATUS = BREAK HIT
 *PC=01E6  A=7 B=6 X=024 Y=026 F=.... SP=4D
 *RUN TIME=TIMEOVER

#HB↵
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 2415  0423 83A LD   Y,3A  7 6 056 03A 0010
 2416  0424 CF1 OR   MY,01 7 6 056 03A 0000 R03A=0 W03A=1
 2417  0425 FDF RET        7 6 056 03A 0000 R01D=6 R01E=6 R01F=1
   :     :   :   :         : :  :   :    :
 2432  0174 FF8 HALT       7 6 024 026 1000
 2433                                      W01F=1 W01E=7 W01D=5 INT1
 2434                                                          INT2
 2435 * 0108 0E6 JP   E6    7 6 024 026 0000
```

..... ブレーク後、HBを実行すると最終ヒストリポインタより
さかのぼって21命令分表示します。

HISTORY DATA DISPLAY BACKWARD/FORWARD HB, HG

Format

#HB↵

#HG↵

Examples

```
#HPS,200↵

#HG↵                    ..... ヒストリポインタ200から21命令分表示します。
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 0200  0128 FD0 POP  A     F 0 020 021 0011 R01F=0
 0201  0129 F70 DEC  M0    0 0 020 021 0010 R000=1 W000=0
 0202  012A 722 JP   NZ,22 0 0 020 021 0010
 0203  012B F71 DEC  M1    0 0 020 021 0000 R001=2 W001=1
   :     :   :   :         : :  :   :    :
 0218  000F C1F ADD  B,0F  F E 013 011 0001
 0219  0010 70E JP   NZ,0E F D 013 011 0001
 0220  000E EE8 LDPX MX,A  F D 013 011 0001 W013=F

#HPS,200↵

#HB↵                    ..... ヒストリポインタ200からさかのぼって21命令分表示します。
  LDC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 0180  000F C1F ADD  B,0F  F 6 03B 021 0001
 0181  0010 70E JP   NZ,0E F 5 03B 021 0001
 0182  000E EE8 LDPX MX,A  F 5 03B 021 0001 W03B=F
 0183  000F C1F ADD  B,0F  F 5 03C 021 0001
   :     :   :   :         : :  :   :    :
 0198  0012 E38 LD   MY,08 F 0 020 021 0011 W021=8
 0199  0013 FDF RET        F 0 020 021 0011 R01C=8 R01D=2 R01E=1
 0200  0128 FDO POP  A     F 0 020 021 0011 R01F=0

#HG↵
  LDC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 2418  0166 B3A LD   Y,3A  7 6 03A 03A 0000
 2419  0167 CAE AND  MX,0E 7 6 03A 03A 0010 R03A=1 W03A=0
 2420  0168 BFE LD   X,2E  7 6 02E 03A 0010
 2421  0169 E20 LD   MX,00 7 6 02E 03A 0010 W02E=0
 2422  016A BF0 LD   X,20  7 6 020 03A 0010
 2423  016B 980 LBPX MX,B0 7 6 021 03A 0010 W020=0 W021=8
 2424  016C 9C1 LBPX MX,C1 7 6 023 03A 0010 W022=1 W023=C
                        ..... ESC入力によりコマンドを終了します。
#
```

HS, HSR, HSW *HISTORY SEARCH PC/MEMORY READ/MEMORY WRITE*

Format

#HS,<アドレス>↵
#HSR,<アドレス>↵
#HSW,<アドレス>↵

Function

ヒストリ情報を以下の条件で検索し、表示します。

(1) HS … <アドレス>で指定されたPCアドレスのヒストリ情報を表示します。

(2) HSR … <アドレス>で指定されたメモリをリードしたヒストリ情報を表示します。

(3) HSW … <アドレス>で指定されたメモリをライトしたヒストリ情報を表示します。

Examples

```
#HS,0700↵                ..... PC=700のヒストリ情報を捜し表示します。
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 1980  0700 FC1 PUSH B     0 0 0FE 0FF 1111 W0F0=0
 2038  0700 FC1 PUSH B     5 1 0FE 0F0 1001 W0FE=1
   :
   :

#HSR,30↵                 ..... 30番地をリードしたヒストリ情報を捜し表示します。
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 0820  0640 EC2 LD   A,MX  0 0 030 0FF 1111 R030=0
 0950  084F EC6 LD   B,MY  0 F 030 0FF 1111 R030=F
   :
   :

#HSW,30↵                 ..... 30番地をライトしたヒストリ情報を捜し表示します。
  LOC    PC  IR OP   OPR.  A B   X   Y IDZC MEMORY OPERATION  OTHER
 0838  0650 E60 LDPX MX,0  0 0 030 0FF 1111 W030=0
 0950  084F E71 LDPY MY,1  0 0 0FF 030 1111 W030=1
   :
   :

#
```

HISTORY POINTER DISPLAY/SET HP, HPS

Format

#HP↵

#HPS,<ヒストリポインタ>↵

Function

ヒストリポインタの表示、変更を行います。

(1) HP … カレントヒストリポインタの値を表示します。

(2) HPS … <ヒストリポインタ>で示す値をカレントヒストリポインタにセットします。最終ヒストリポインタを越える値が入力された場合は最終ヒストリポインタ値をカレントヒストリポインタにセットします。

(3) ヒストリポインタは10進4桁で表示、指定します。

Examples

```
#HP↵
 * LOC=2058  ..... ブレーク時のヒストリポインタ(最終値)を表示します。
#HPS,1000↵  ..... ヒストリポインタを1000にセットします。
#HP↵
 * LOC=1000  ..... カレントヒストリポインタは1000になっています。
#HPS,9999↵
 * LOC=2058  ..... ポインタを最終に戻します。最終値以上を越えた指定は最終値とみなします。
#HP↵
 * LOC=2058
#
```

CHECK ICE HARDWARE CHK

Format

#CHK↵

Function

ICEが行ったイニシャルテストの結果を表示します。
(ICEは電源投入時にイニシャルテストを実行します。)

下記項目がイニシャルテストとして行われます。

(1) ICEのファームウェアのサムチェックテスト

(2) ICEのRAMのR/Wテスト

Examples

```
#CHK↵
 * ROM CHECK ERROR 5F=>FF *
 * RAM CHECK ERROR 001111 55=>FF *
```

テストでエラーが検出された場合エラーメッセージを表示します。

```
#CHK↵
#              ..... 正常時はコマンド待ちになります。
```

Note

エラーメッセージが表示された場合は、ハードの故障と考えられますので以後の使用は避けてください。

DXY *DISPLAY X, Y REGISTER & MX, MY CONTENT*

Format

#DXY↵

Function

Xレジスタ(XP、XH、XL)とYレジスタ(YP、YH、YL)およびMX、MY(X、Yで指定されているメモリ)の現在の内容を表示します。

Examples

```
#DXY↵
 X=070  MX=  5
 Y=07C  MY=  F

#DXY↵
 X=200  MX=-:OV ..... RAM領域を越えていることを示します。読み出し動作は行いません。
 Y=050  MY=-:   ..... ライトオンリーエリアを示します。読み出し動作は行いません。

#DXY↵
 X=E73  MX=  /  ..... E73が未使用領域であることを示します。
 Y=252  MY=  F  ..... 読み出し動作は行いません。

#
```

CVD, CVR *DISPLAY/RESET COVERAGE*

Format

#CVD,<アドレス1>,<アドレス2>↵
#CVD↵
#CVR↵

Function

カバレッジ情報の表示とクリアを行います。

(1) CVD … <アドレス1>から<アドレス2>の範囲のカバレッジ情報を表示します。
アドレスを省略した場合、すべてのカバレッジ情報を表示します。

(2) CVR … カバレッジ情報をクリアします。

Examples

```
#CVD,100,110↵          ..... 100から110番地までのカバレッジ情報を表示します。
 *CV 0100
 *CV 0109..0110
#

#CVD↵                  ..... すべてのカバレッジ情報を表示します。
 *CV 0100
 *CV 0109..02FF
 *CV 0400..04FF
#

#CVR↵                  ..... カバレッジ情報をクリアします。
#
```

2.4.2 SETコマンド群

A	ASSEMBLE PROGRAM	VIII-36
FP	FILL PROGRAM	VIII-37
FD	FILL DATA RAM	VIII-38
MP	MOVE PROGRAM	VIII-39
MD	MOVE DATA RAM	VIII-40
SP	SET PROGRAM	VIII-41
SD	SET DATA RAM	VIII-42
SR	SET REGISTER	VIII-43
SXY	SET MX, MY DATA	VIII-44
HC	SET HISTORY CONDITION	VIII-44
HA	SET HISTORY RANGE	VIII-45
HAD	DISPLAY HISTORY RANGE	VIII-45
HAR	RESET HISTORY RANGE	VIII-45

A *ASSEMBLE PROGRAM*

Format

#A,<アドレス>↵ **(ガイダンス付)**

Function

命令ニーモニックをアセンブルして<アドレス>で示される番地から格納します。

(1) S1C62 Familyの命令一覧表のニーモニック、オペランドをサポートします。

(2) オペランドの表現は下記形式となります。

p … 00～03の値
s … 00～FFの値
l … 00～FFの値
i … 00～0Fの値
r、q … A、B、MX、MYのいずれか

- 原則として16進表現で最後にHをつけない表現とします。
- データ桁は0で始まる場合3桁まで入力可とします。
  0FF 入力 … FFとみなし有効です。
  00FF入力 … エラーとします。
- p、s、l、iの入力時に有効でない値はエラーとします。
- i入力部のみ2進表現(xxxxB)入力を許可します。 ここでxは0または1の4桁固定長とし、最後にBを入力します。3桁以下の入力時はエラーまたは16進表現として扱います。

(3) 大文字でも小文字でも入力可能です。

(4) ニーモニックとオペランドの間は1文字以上のスペースまたはタブコードとします。

(5) サポートされない命令を入力したときはエラーとします。

(6) A、B入力はレジスタ優先となります。イミディエート値指定時は0A、0Bと入力します。
`LD A,B` … Bレジスタ内容がAレジスタにロードされます。
`LD B,0A` … イミディエート値AがBレジスタにロードされます。

Examples

```
#A,100↵
 0100 LD A,0F↵          ..... コマンドをキー入力します。
                              アドレスを表示し、ニーモニック入力待ちとなります。
                              命令ニーモニック、オペランドを入力します。
 0101 /↵                ..... /↵キー入力によりコマンドを終了します。

#A,200↵
 0200 PUSH XP↵          ..... S1C62XXX で許されないニーモニック入力があれば
     * ERROR *                エラーとなり、再度同一アドレスを表示してニーモニック
 0200 NOP5↵                   要求を行います。
 0201 JJJ 0FF↵
     * ERROR *
 0201 LD A,FF↵          ..... オペランドの有効範囲を越えているためエラーとなります。
     * ERROR *
 0201 LD A,0F↵
 0202 /↵

#A,202↵
 0202 ^↵                ..... ^キーでひとつ前のアドレスに戻ります。
 0201 /↵
#
```

Note

ESCキーは受け付けません。終了は/↵キーの入力で行います。

FILL PROGRAM FP

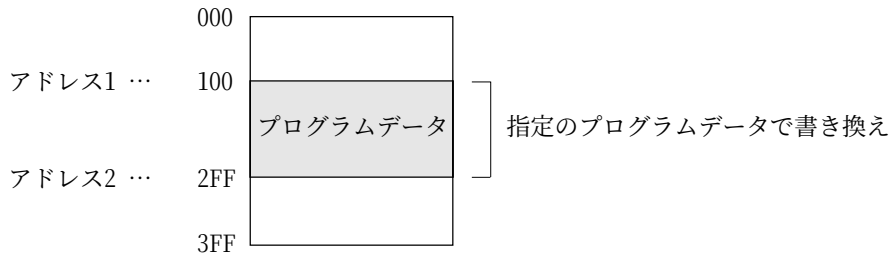
Format

#FP,<アドレス1>,<アドレス2>,<プログラムデータ>↵

Function

プログラム領域(ICEエミュレーションメモリ)の<アドレス1>番地から<アドレス2>番地までの内容を<プログラムデータ>に書き換えます。

プログラム領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Examples

```
#FP,0,3FF,FFB↵          ..... プログラム領域000～3FF番地の内容をFFB(NOPコード)に
                              書き換えます。

#FP,100,200,FF9↵        ..... 未定義コード検出時はエラーを表示し、コマンドは実行
 * COMMAND ERROR *            されません。

#FP,200,100,FFF↵
 * COMMAND ERROR *      ..... アドレス1>アドレス2のためエラーとなります。

#FP,200,200,FFF↵        ..... 200番地を命令コードFFF(NOP7)に変更します。
                              正常終了します。
#
```

FD *FILL DATA RAM*

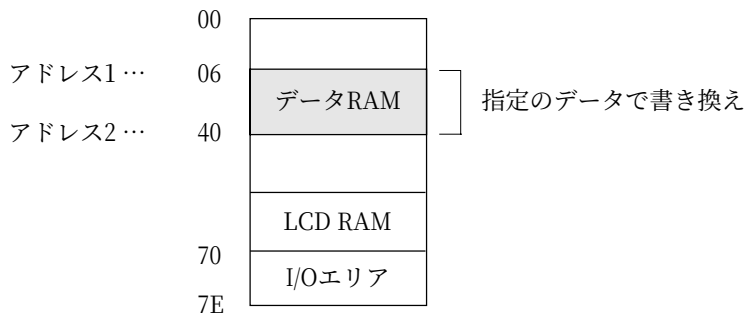
Format

#FD,<アドレス1>,<アドレス2>,<データ>↵

Function

データRAM領域の<アドレス1>番地から<アドレス2>番地までの内容を<データ>で書き換えます。<データ>は16進表現と2進表現が受け付けられます。

データRAM領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Examples

```
#FD,60,7E,A↵          ..... データRAM60番地～7E番地の内容をAに書き換えます。

#FD,10,2F,0101B↵      ..... 10番地～2F番地をデータ0101(2進) =5(16進)で書き換えます。

#FD,50,1FF,0↵
 * COMMAND ERROR *    ..... RAM領域(S1C6S3N7/6S3B7/6S3L7の場合7Eまで)を越えた
                            指定のためエラーとなりコマンドは実行されません。
#FD,70,60,0↵
 * COMMAND ERROR *    ..... アドレス1>アドレス2のためエラーとなります。

#FD,0,7E,B↵           ..... RAM全エリアを(S1C6S3N7/6S3B7/6S3L7の場合)データB(16進)で書き換えます。
#FD,40,40,0↵          ..... 40番地の内容を0で書き換えます。

#
```

Notes

(1) 2進表現の場合、4桁の0(または1)とBの入力(全部で5文字)のみ受け付けます。

(2) I/Oエリアでリードオンリー指定のアドレスには書き込み動作は行いません。

(3) 指定した番地に未使用領域がある場合、未使用領域を除いてデータを書き換えます。

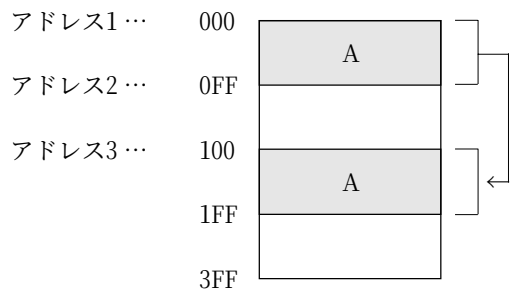
Format

#MP,<アドレス1>,<アドレス2>,<アドレス3>↵

Function

プログラム領域(ICEエミュレーションメモリ)の<アドレス1>番地から<アドレス2>番地までの内容を<アドレス3>番地以降に転送します。

プログラム領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Examples

```
#MP,0,FF,100↵          ..... プログラム領域000番地～0FF番地の内容を100番地～1FF番地
                             の領域に転送します。
#MP,100,2FF,300↵
 * COMMAND ERROR *     ..... 転送エリアがプログラム領域(S1C6S3N7/6S3B7/6S3L7の場合
                             3FF)を越える場合はエラーが表示されコマンドは実行されま
                             せん。
#MP,200,100,300↵
 * COMMAND ERROR *     ..... アドレス1>アドレス2のためエラーとなります。

#MP,200,200,300↵       ..... 200番地の内容を300番地にコピーします。
                             正常に実行されます。
#
```

MD *MOVE DATA RAM*

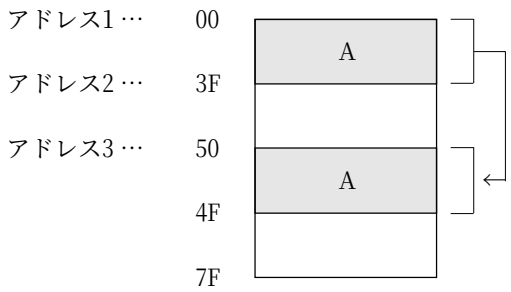
Format

#MD,<アドレス1>,<アドレス2>,<アドレス3>↵

Function

データRAM領域の<アドレス1>番地から<アドレス2>番地までの内容を<アドレス3>番地以降に転送します。

データRAM領域 (S1C6S3N7/6S3B7/6S3L7の場合)



Examples

```
#MD,10,1F,30↵          ..... データRAM10番地～1F番地の内容を30番地～3F番の領域
                             に転送します。
#MD,00,3F,70↵
 * COMMAND ERROR *     ..... 転送エリアがデータRAM領域(S1C6S3N7/6S3B7/6S3L7の場
                             合7F)を越える場合はエラーが表示され、コマンドは実行されま
                             せん。
#MD,30,20,50↵
 * COMMAND ERROR *     ..... アドレス1>アドレス2のためエラーとなります。

#MD,30,30,50↵          ..... 30番地の内容を50番地にコピーします。
                             正常に実行されます。
#MD,E00,E1F,E60↵
 * UNUSED AREA *       ..... 転送エリア(転送先、転送元のいずれか)に未使用領域がある
                             場合は未使用領域エラーが表示されます。
#                            (S1C6S460の場合)
```

Notes

(1) 転送先にI/Oエリアのリードオンリー領域がある場合、その領域への書き込み動作は行われません。

(2) 転送元にI/Oエリアのライトオンリー領域がある場合、その領域の読み出し動作は行われません。この場合、転送先には"0"が書き込まれます。

(3) 転送アドレスがI/Oエリアの"リード可能ビットとライトオンリービット混在アドレス"の場合は読み出しまたは書き込み動作が行われます。

Format

#SP,<アドレス>↵ **(ガイダンス付)**

Function

プログラム領域の<アドレス>番地からの内容を表示または変更します。

Examples

```
#SP,100↵
 0100 FFF:↵              ..... 100番地の内容が読み出されます。↵入力のみでは変更
                               されません。
 0101 FFF:FFB↵           ..... 新たなデータを書き込みます。
 0102 FFF:FF9↵
 * CODE ERROR *          ..... 未定義コードが検出された場合、エラーを表示します。
 0102 FFF:F05↵                 この場合内容は書き変わらず同一アドレスを表示します。
 0103 FFF:A6B↵
 0104 FFF:^↵             ..... ^↵入力で1つ前のアドレス表示に戻ります。
 0103 A6B:^↵
 0102 F05:F06↵
 0103 A6B:↵
 0104 FFF:ABx↵
 * COMMAND ERROR *       ..... データ指定エラーのためエラーを表示します。
 0104 FFF:ABC↵
 0105 FFF:/↵             ..... /↵入力でコマンドを終了します。

#SP,400↵
 * COMMAND ERROR *       ..... プログラム領域(S1C6S3N7/6S3B7/6S3L7の場合3FF)を越え
                               るためエラーとなります。
#SP,3FE↵
 3FE FFF:011↵
 3FF FFF:FFB↵            ..... 最終番地入力後はコマンドを終了します。

#
```

SD *SET DATA RAM*

Format

#SD,<アドレス>↵ **(ガイダンス付)**

Function

データRAM領域の<アドレス>番地からの内容を表示または変更します。

(1) リードオンリーエリアへの書き込み動作は行いません。

(2) ライトオンリーエリアの読み出し動作は行いません。

Examples

```
#SD,20↵
 20 5:A↵                ..... 20番地をAに変更します。
 21 5:^↵                ..... ^↵で1つ前のアドレスに戻ります。
 20 A:B↵
 21 5:F↵
 22 5:/↵                ..... /↵でコマンドを終了します。

#SD,FFF↵
 * COMMAND ERROR *      ..... RAMエリアの最大値(S1C6S3N7/6S3B7/6S3L7の場合7F)を
                              越える指定はエラーとなります。
#SD,70↵
 70 4:-↵                ..... リードオンリーアドレスのためハイフン表示のみを行い、
                              データ入力はできません。
 71 F:-↵
 72 5:-↵
 73 6:-↵
 74 6:5↵
 75 8:4↵
 76 5:A↵
 77 8:9↵
 78 8:5↵
 79 A:-↵                ..... リードオンリーアドレスのためハイフン表示のみを行い、
                              データ入力はできません。
 7A B:-↵
 :  :  :
 7E F:-↵                ..... 最終アドレスまで入力するとコマンドを終了します。

#SD,E50↵
 * UNUSED AREA *        ..... 未使用領域を指定した場合、* UNUSED AREA *と
                              表示します。 (S1C6S460の場合)
#SD,ECE↵
 ECE 0:F↵
 ECF 4:F↵
 * UNUSED AREA *        ..... データ設定中に未使用領域に入った場合、
                              * UNUSED AREA *と表示します。 (S1C6S460の場合)
#
```

Format

#SR↵ **(ガイダンス付)**

#SP,<レジスタ名>,<データ>↵

Function

Evaluation Board CPUのレジスタ群の表示および変更を行います。

(1) <レジスタ名>を指定した場合は、そのレジスタに<データ>をセットします。

(2) <レジスタ名>にはPC、A、B、X、Y、FI、FD、FZ、FC、SPが指定できます。

Examples

```
#SR↵
 PC=0100:0105↵          ..... 変更したいレジスタに対しデータと↵を入力します。
  A=    5:↵                   変更しないレジスタは↵入力のみで次に進めます。
  B=    A:5↵
  X= 02F:20↵
  Y= 010:1A↵
 FI=    0:1↵
 FD=    1:↵
 FZ=    0:↵
 FC=    1:0↵
 SP=   4F:^↵            ..... ^↵で1つ前の設定に戻します。
 FC=    0:1↵
 SP=   4F:↵

#SR,X,AA↵               ..... Xレジスタのみ0AAHに変更します。

#SR↵
 PC= 105:↵              ..... ↵入力で現在の値を保持します。
  A=    5:↵
  B=    5:↵
  X=   2A:↵
  Y=   2A:↵
     :
 SP=   4F:↵

#
```

Note

/↵でコマンドは終了しません。↵で最終レジスタまで進めてください。

SXY *SET MX, MY DATA*

Format

#SXY↵ **(ガイダンス付)**

Function

X(XP、XH、XL)レジスタ、Y(YP、YH、YL)レジスタとMX、MY(X、Yで示されるメモリ)の内容を表示します。また、MX、MYの内容を変更します。

Examples

```
#SXY↵                          ..... 表示のみを行う場合は↵入力のみで進めます。
 X=040 MX=5:↵
 Y=030 MY=A:↵

#SXY↵
 X=040 MX=5:0↵                 ..... MX ┐
 Y=030 MY=A:F↵                  ..... MY ┘ に新たなデータを設定します。

#SXY↵
 X=070 MX=3:-                   ..... リードオンリーエリアには入力できません。
 Y=FFF MY=-:OV                  ..... RAMサイズを越えている場合、入力はできません。

#SXY↵
 X=E52  MX * UNUSED AREA *      ..... E52は未使用領域エラーが表示されます。
 Y=1A7  MY=1:3↵                       (S1C6S460の場合)

#
```

HC *SET HISTORY CONDITION*

Format

#HC,S/C/E↵

Function

ヒストリ採取領域の設定をブレークポイントによって行います。
ブレークポイントには、"[ ]"の表示が付きます。

Examples

```
#HC,S↵                         ..... ブレークポイントからヒストリを採取します。

#HC,C↵                         ..... ブレークポイントの前後でヒストリを採取します。

#HC,E↵                         ..... ブレークポイントまでのヒストリを採取します。
                                      (デフォルト値)
#
```

SET/DISPLAY/RESET HISTORY RANGE HA, HAD, HAR

Format

#HA,<アドレス1>,<アドレス2>/ALL↵
#HAD↵
#HAR,<アドレス1>,<アドレス2>/ALL↵

Function

ヒストリー採取領域のPCアドレスの設定、表示、解除を行います。

(1) HA … <アドレス>で指定された範囲を採取します。ALLを指定すると全アドレスを指定したことになります。

(2) HAD … ヒストリ採取領域のアドレスを表示します。

(3) HAR … <アドレス>で指定された範囲の採取を行いません。ALLを指定すると、ヒストリの採取を行いません。

Examples

```
#HAR,ALL↵                ..... ヒストリ採取領域の全解除。

#HA,300,400↵             ..... ヒストリ採取領域を指定します。

#HA,100,200↵

#HA,500,500↵

#HAD↵                    ..... ヒストリ採取領域を表示させます。
 * HA 0100..0200
 * HA 0300..0400
 * HA 0500

#
```

2.4.3 BREAK & GOコマンド群

BA SET BREAK ADDRESS CONDITION VIII-48
BAR RESET BREAK ADDRESS CONDITION VIII-48
BD SET BREAK DATA CONDITION VIII-49
BDR RESET BREAK DATA CONDITION VIII-49
BR SET BREAK REGISTER CONDITION VIII-50
BRR RESET BREAK REGISTER CONDITION VIII-50
BM SET BREAK MULTIPLE CONDITION VIII-52
BMR RESET BREAK MULTIPLE CONDITION VIII-52
BC BREAK CONDITION DISPLAY VIII-54
BRES RESET ALL BREAK CONDITION VIII-55
G GO TARGET PROGRAM VIII-56
T SINGLE STEP TRACE VIII-58
U SINGLE STEP TRACE & LAST INFORMATION DISPLAY VIII-60
BE BREAK ENABLE MODE SET VIII-61
BSYN BREAK DISABLE & SYNC MODE SET VIII-61
BT BREAK TRACE MODE SET VIII-61
BRKSEL BREAK ADDRESS MODE SELECT VIII-62

BA, BAR *SET/RESET BREAK ADDRESS CONDITION*

Format

#BA,<アドレス1>,<アドレス2>,<アドレス3>,<アドレス4>↵
#BAR,<アドレス1>,<アドレス2>,<アドレス3>,<アドレス4>↵

Function

PCに関するブレーク条件を設定します。

(1) BA … <アドレス>で示される値をブレーク条件に設定します。
複数アドレスにセットする場合カンマで区切り入力します。
連続アドレスの設定は"."(ピリオド)2個で入力します。
<アドレス3>..<アドレス4>…<アドレス3>≦PC≦<アドレス4>がブレーク条件となります。

(2) BAR … BAで設定した条件を個別にクリアします。

(3) BA、BARで1回のコマンドで入力できるアドレスは1行以内(80桁)であれば何個でも設定できます。それ以上の場合は2回以上に分けて入力してください。

(4) BAコマンドを数回実行したとき、過去の設定も有効となります。

(5) BMコマンドを実行するとBA条件はすべて解除されます。

(6) ブレーク時にBA条件がクリアされるモードと条件が残るモードがあります。
(BRKSELコマンドVIII-62を参照してください。)

Examples

```
#BA,100,200,101,1FF↵      ..... 100、200、101、1FF番地にブレーク条件を設定します。

#BA,300..3FF↵             ..... 300～3FF番地にブレーク条件を設定します。

#BAR,100,200..3FF↵        ..... 100番地、200～3FF番地のブレーク条件を解除します。
                                (201～2FF番地はブレーク条件が指定されていませんが
                                BAR指定を行ってもエラーにはなりません。)
#BC↵
 BA 0201                  ..... BCコマンド(VIII-54を参照)によりBA条件が表示されます。
 BA 02FF
 BD NONE
 BR NONE
  :
  :

#
```

SET/RESET BREAK DATA CONDITION BD, BDR

Format

#BD↵ **(ガイダンス付)**

#BDR↵

Function

データRAMのリードライトに関するブレーク条件を設定します。

(1) BD … データRAMのアドレス、データ、R/Wについてブレーク条件を指定します。アドレスについては1ポイント、データについては0～F、またはビット単位のマスク、R/Wについてはリード、ライト、マスクの指定が可能です。アドレスデータ、R/Wの3条件が共に一致したときにブレークします。

(2) BDR … BDで指定した条件を解除します。

(3) BDではブレーク条件は1ポイントのみとなります。ただし、BA、BR指定との混在は可能です。

(4) BMコマンド実行によりBD条件は解除されます。

Examples

```
#BD↵
 ADDR ---:074↵          ..... BD条件が存在しない場合は"-"(ハイフン)表示となります。
 DATA   -:5↵                  アドレスに74、データに5、 R/Wにマスク*を指定します。
 R/W    -:*↵
```

以上により、データRAMの74番地にデータ5をライトまたは、74番地からデータ5をリードしたときにブレークする設定となります。

```
#BD↵
 ADDR 074:↵             ..... 設定変更なしのときは↵で次の設定に進みます。
 DATA  5 :*↵            ..... データ指定をマスクします。
 R/W   * :W↵            ..... R/Wをライト指定します。
```

本設定ではデータRAMの74番地にライト動作を行ったときにブレークします。

```
#BD↵
 ADDR 074:↵
 DATA  5 :1*1*B↵
 R/W   W :↵
```

本設定ではデータRAMの74番地に2^3ビットと2^1ビットが1のライト動作を行ったときにブレークします。

```
#BDR↵                   ..... BD条件をクリアします。

#BD↵
 ADDR ---:↵             ..... BD指定が解除されている場合の↵入力は解除のままとなります。
#
```

BR, BRR *SET/RESET BREAK REGISTER CONDITION*

Format

#BR↵ **(ガイダンス付)**

#BRR↵

Function

Evaluation Board CPUのレジスタ(A、B、FLAG、X、Y)に関するブレーク条件を設定します。

(1) BR … レジスタA、B、FLAG、X(XP、XH、XL)、Y(YP、YH、YL)についてブレーク条件を設定します。各レジスタはブレーク条件マスクが可能です(マスクされたレジスタは値が何であっても他のレジスタの条件でブレークします)。CPU内の各レジスタと指定した各レジスタ値が等しいときブレークします。

(2) BRR … BRで指定した条件を解除します。

(3) BRではブレーク指定は1ポイントとなります。ただし、BA、BD指定との混在は可能です。

(4) BMコマンド実行によりBR条件は解除されます。

Examples

```
#BR↵
 A       -:C↵          ..... BR条件が設定されていない場合"-"(ハイフン)表示となります。
 B       -:*↵                ブレーク条件を順次設定します。
 FI      -:1↵
 FD      -:*↵          ..... マスクする場合*をキー入力します。これによりFDの値に
 FZ      -:0↵                関係なくブレークします。
 FC      -:*↵
 X     ---:040↵
 Y     ---:^↵          ..... 設定ミスがあった場合、^↵で1つ前の設定に戻します。
 X     ---:041↵
 Y     ---:030↵
```

以上の設定によりブレーク条件は"A=C、かつFI=1、かつFZ=0、かつX=41、かつY=30になったとき"という指定になります。

```
#BR↵
 A       C:↵           ..... 前に設定したブレーク条件が読み出されます。
 B       *:↵                 変更しない場合↵入力で次の設定に進みます。
 FI      1:*↵
 FD      *:↵
 FZ      0:*↵
 FC      *:↵
 X     041:042↵
 Y     030:*↵
```

以上でA=C、かつX=42のブレーク条件が設定されます。

```
#BRR↵                 ..... BRRでBR条件はクリアされます。
#BR↵
 A       -:↵           ..... BR指定が解除されている場合の↵入力は解除のままとなります。
#
```

SET/RESET BREAK REGISTER CONDITION BR, BRR

Format

#BR↲ **(ガイダンス付)**

#BRR↲

Examples

```
#BR↲
 A       -:0↲
 B       -:0↲
 FI      -:*↲
 FD      -:*↲
 FZ      -:*↲
 FC      -:*↲
 X     ---:40↲
 Y     ---:30↲
```

ブレーク条件をA=0、B=0、X=40、Y=30に設定します。

```
#BR↲
 A       0:↲
 B       0:5↲
 FI      *:/↲          ..... /↲入力で以後の設定変更なしにコマンドを終了します。
```

ブレーク条件はA=0、B=5、X=40、Y=30となります。

```
#
```

Notes

(1) BR指定後GOを行ってもターゲットのリアルタイム動作は保証されます。

(2) 各機種によりRAM領域は異なりますが、BMコマンド内のX、Yは各機種共FFFまで設定可能です。

BM, BMR *SET/RESET BREAK MULTIPLE CONDITION*

Format

#BM↵ **(ガイダンス付)**

#BMR↵

Function

Evaluation Board CPUのPC、データRAMアクセス、レジスタ値のすべての条件が一致したときブレークさせる複合ブレーク機能を指定します。

(1) BA、BD、BR(VIII-48～VIII-51参照)指定はそれぞれが独立に指定可能ですがBM条件指定ではPC、データRAMアクセス、レジスタ値すべてがブレーク条件と一致したときにブレークします。すなわちBA、BD、BRのAND指定と考えることができます。

(2) BMの指定により以前設定されていたBA、BD、BR条件は解除されます。
また、BMの設定後BA、BD、またはBR設定が行われるとBM設定は解除されます。

(3) BMRコマンドはBMの設定を解除します。

(4) BMではブレーク条件は1ポイントのみとなります。ただし、各レジスタ設定についてのマスクは可能です。

Examples

```
#BM↵
 PC    ----:100↵        ..... BM条件が解除されているときは"-"表示となります。
 ADDR  ---:70↵                PC=100、RAMアドレス=70、RAMデータ=A、Dフラグ、
 DATA    -:A↵                 Cフラグ=1、Yレジスタ= 3Eでブレーク条件を指定します。
 R/W     -:*↵                 100番地の命令実行中にRAMの70番地にアクセスを行い、
 A       -:*↵                 かつリード/ライトしたデータがA、FD、FCがセット、
 B       -:*↵                 Yレジスタが3Eの場合にブレークします。
 FI      -:*↵                 (プログラムループ時のブレーク設定に有効です。)
 FD      -:1↵
 FZ      -:*↵
 FC      -:1↵
 X      ---:*↵        ..... ブレーク条件に加えない所は*によりマスクします。
 Y      ---:3E↵
```

SET/RESET BREAK MULTIPLE CONDITION BM, BMR

Format

#BM↵ **(ガイダンス付)**

#BMR↵

Examples

```
#BM↵
 PC     100:*↵          ..... PCについてマスクします。
 ADDR    70:71↵
 DATA     A:^↵          ..... ^↵で1つ前の設定に戻ります。
 ADDR    71:72↵
 DATA     A:↵           ..... ↵のみで前の指定のままになります。
 R/W      *:W↵
 A        *:↵
 B        *:↵
 FI       *:↵
 FD       1:↵
 FZ       *:↵
 FC       1:↵
 X        *:70↵
 Y       7E:↵
```

以上により、RAMアドレス72にデータAをライトした場合で、レジスタX=70、Y=7E、FD=1、FC=1のときにブレークします。

```
#BM↵
 PC       *:100↵
 ADDR    71:/↵          ..... /↵で以下の設定を変更しません。上記条件にPC=100が
                              加わることになります。

#BMR↵                   ..... BMの設定条件を解除します。

#BM↵
 PC     ----:↵          ..... BM指定が解除されている場合の↵入力は解除のままと
                              なります。
#
```

Notes

(1) BMコマンドを使用した場合にはBA、BD、BR指定が自動的に解除されます。

(2) 本コマンドはメモリアクセスを行う命令実行時のみブレーク比較を行います。
ADDR、データ、R/Wをマスクしても上記制限はそのまま残ります。したがって、PC、レジスタ値が一致しても、データメモリアクセスが存在しない命令ではブレークしません。

(3) 各機種によりRAM領域は異なりますが、BMコマンド内のX、Yは各機種共FFFまで設定可能です。

BC *BREAK CONDITION DISPLAY*

Format

#BC↵

Function

現在のブレーク条件を表示します。

Examples

```
#BC↵                          ..... 電源投入後、ブレーク条件の確認をします。
 * BA NONE                          すべてのブレーク条件は解除されています。
 * BD NONE
 * BR NONE
 * BM NONE
 * BREAK ENABLE MODE  ..... ブレークイネーブルモード ┐
 * BREAK STOP MODE    ..... ブレークストップモード   ┘ となっています。
 * TIME COUNT MODE    ..... 実時間測定モード

#BA,100,101↵

#BC↵                          ..... アドレスブレーク条件設定後の確認を行います。
 * BA 0100..0101
 * BD NONE
 * BR NONE
 * BM NONE
 * BREAK ENABLE MODE
 * BREAK STOP MODE
 * TIME COUNT MODE

#BRES↵

#BA,100,102↵

#BC↵
 * BA 0100                    ┐ アドレスが連続しない場合、BA条件は複数行に分けて
 * BA 0102                    ┘ 表示されます。
  :
  :

#
```

RESET ALL BREAK CONDITION BRES

Format

#BRES↵

Function

すべてのブレーク条件(BA指定、BD指定、BR指定またはBM指定)を解除します。

Examples

```
#BRES↵

#BC↵
 * BA NONE
 * BD NONE
 * BR NONE
 * BM NONE
 * BREAK ENABLE MODE
 * BREAK STOP MODE
 * TIME COUNT MODE

#
```

Note

ブレーク条件は解除しますが、ブレークモード(イネーブル/ディセーブル、トレース/ストップ、タイム/ステップ)は変更しません。

G　　*GO TARGET PROGRAM*

Format

#G↵
#G,<アドレス>↵
#G,R↵

Function

ターゲットプログラムを実行させます。ブレーク条件が検出されると実行を停止しブレークステータスを表示してコマンドを終了します。

■ 開始アドレスの指定

(1) <アドレス>入力がある場合そのアドレスから実行します。

(2) R指定がある場合Evaluation Board CPUをリセットし、リセットアドレス0100番地から実行します。

(3) <アドレス>、R指定共に省略した場合、現在のアドレス(前回、ブレークしたときにステータス表示されたPC)からスタートします。
電源投入後G↵入力があったときには0100番地から実行しますが、この場合はEvaluation Board CPUはリセットしません。

■ ブレークモードとブレーク条件

項番	ブレークモード(注)	ブレーク条件	備　考
1	ブレークイネーブル &ブレークストップ モード	・リセットスイッチ ・ブレークスイッチ ・ブレーク指定コマンド (BA, BD, BR, BM) ・ESC入力	電源投入時のモードです。
2	ブレークイネーブル &ブレークトレース モード	・リセットスイッチ ・ブレークスイッチ ・ESC入力	ブレーク指定コマンドの条件と Evaluation Board CPUの実行サイクル が一致するとブレークステータス のみ表示しGOを再開します。
3	BSYNモード &ブレークストップ モード	・リセットスイッチ ・ブレークスイッチ ・ESC入力	ブレーク指定の条件とEvaluation Board CPUの実行サイクルが一致すると SYNC端子にパルスを出力します。

注: ブレークモードについては2.3.2項を参照してください。

■ GO中の表示

項番	表示モード(注)	表示方法	
1	オンザフライ 表示モード	#G↵ *PC=xxxx	PCを約500msecごとにサンプリングして表示します。HALT中はHALTと表示します。 電源投入時本モードに設定されます。
2	オンザフライ 禁止モード	#G↵	実行状態は表示されません。

注: 表示モードについては2.3.4項を参照してください。

Format

#G↵

#G,<アドレス>↵

#G,R↵

Function

■ ブレーク時の表示

```
#G↵
  *PC=xxxx
  *EMULATION END STATUS = BREAK HIT          ..... ①
  *PC=0100 A=0 B=0 X=70 Y=00 F=ID.C SP=10    ..... ②
  *RUN TIME = xxx mS                         ..... ③
```

ブレークステータスの表示を行います。

①部には BREAK HIT、ESC KEY、BREAK SW が表示されます。
リセットスイッチが押された場合はブレークステータス表示は行われず、
`* ICE6200 RESET SW TARGET *`を表示してコマンド待ちになります。

②部には停止したPC(次に実行するアドレス)と停止したときのレジスタ内容が表示されます。

③部にはTIMコマンドの設定にしたがい実行時間または実行ステップ数が表示されます。
(TIMコマンドについてはVIII-74を参照してください。)

Examples

```
#OTF↵          ..... オンザフライ指定コマンドです。
 * ON THE FLY ON *

#BE↵           ..... ブレークイネーブル指定コマンドです。
 * BREAK ENABLE MODE *

#BT↵           ..... ブレークストップモード指定コマンドです。
 * BREAK STOP MODE *
```

電源投入時は左記の指定になっており、コマンド入力は省略できます。

```
#G,R↵          ..... ターゲットおよびEvaluation Boardをリセットし、 リセット番地
                      (0100番地)から実行します。
 *PC=xxxx      ..... 周期的にPC表示を行います。
 *EMULATION END STATUS = BREAK HIT          ..... ①
 *PC=01FF A=5 B=0 X=70 Y=05 F=..ZC SP=20    ..... ②
 *RUN TIME = 100mS                          ..... ③
```

① ブレーク条件(BA条件01FEセット)によりブレークしたことを示します。

② Fはリセットされたビットを"."、セットされたビットをフラグの頭文字で示します。

③ ラン時間は100mSであったことを示します。

```
#
```

T *SINGLE STEP TRACE*

Format

#T,<アドレス>,<ステップ数>↵
#T,<アドレス>↵
#T,,<ステップ数>↵
#T↵

Function

プログラムのトレース、シングルステップ動作を実行させます。

(1) <アドレス>で示された番地から<ステップ数>で示された回数(10進で65535で指定可)分ターゲットプログラムを実行します。命令実行ごとにPC、命令語、レジスタ内容を表示します。

(2) <ステップ数>を省略した場合は1ステップのみ実行します。

(3) <アドレス>を省略した場合は現在のPC(前回のTコマンド終了時のPC)から<ステップ数>分実行します。

(4) <アドレス>、<ステップ数>共省略した場合は現在のPCから1ステップのみ実行します。電源投入後本指定が行われるとPC=0100として1ステップ実行します。

(5) ステップ数が1(#T,<アドレス>または#T)の場合は1ステップ実行後、コマンド終了とはならず、次の[SP]キー入力によりさらに1ステップ実行します。この場合、[ESC]キー入力によりコマンドは終了します。

(6) 上記(1)においても[ESC]キー入力によりコマンドは終了します。

Examples

```
#T,100,3↵
 *PC=0100 IR=FFF NOP7     A=0  B=0  X=00F  Y=00F  F=IDZC  SP=10
 *PC=0101 IR=E05 LD   A,5 A=5  B=0  X=00F  Y=00F  F=IDZC  SP=10
 *PC=0102 IR=B05 ADC XH,5 A=5  B=0  X=051  Y=00F  F=IDZC  SP=10
```

実行したPCを表示します。　命令コードとニーモニックを表示します。　フラグはセット時にフラグの頭文字をリセット時に"."を表示します。

(3ステップ実行後、現在のPCは0103となっています。)

```
#
```

Format

#T,<アドレス>,<ステップ数>↵
#T,<アドレス>↵
#T,,<ステップ数>↵
#T↵

Examples

#T↵ ..... 現在PC(=103)からシングルステップ動作させ、SPで順次実行させます。

```
 *PC=0103 IR=FDF RET      A=5 B=0 X=04F Y=03F F=IDZC SP=013 ..... SP
 *PC=01AA IR=AD1 OR   A,B A=5 B=0 X=04F Y=03F F=ID.C SP=013 ..... ESC
```

..... ESCでコマンドを終了させます。

```
#T↵
 *PC=01AB IR=xxx PSET   2 A=x B=x X=xxx Y=xxx F=xxxx SP=013
 *PC=01AC IR=xxx JP    10 A=x B=x X=xxx Y=xxx F=xxxx SP=013 ..... ESC

#
```

PSET命令は次の命令と対になって使用されるため、シングルステップ 動作でも一度に2命令実行されます。

```
#T↵
 *PC=01AD IR=xxx HALT _
                      └─ カーソル
#
```

TコマンドでHALT命令を実行すると、ターゲット割り込みが入るまで上記のように命令ニーモニックを表示し、レジスタ値を表示しません。正しく割り込みが入るとレジスタ値を表示し、次のSP待ちとなります。このときのSP入力では割り込みルーチンから再スタートします。
ターゲット割り込みが行われないときはESCによりコマンドを強制終了できます。
この場合はHALT命令は終了し、Tコマンドも終了します。次のTコマンドではHALT命令の次の番地から実行します。

Notes

(1) Tコマンドはリアルタイム性がありません。このためターゲットのタイマー等が実際より多く更新される場合があります。(詳細は2.3.13項を参照してください。)

(2) 本コマンド実行後のHコマンド入力では* NO HISTORY DATA *が表示されます。したがって、ヒストリ解析を行う場合はGコマンドを使用してください。

U SINGLE STEP TRACE & LAST INFORMATION DISPLAY

Format

#U,<アドレス>,<ステップ数>↵

#U,,<ステップ数>↵

Function

プログラムのトレース、シングルステップ動作を行い、最終結果のみ表示します。

(1) <アドレス>で示された番地から<ステップ数>で示された回数(10進で65535まで指定可)分ターゲットプログラムを実行しますが、実行結果は最後の命令しか表示しません。

(2) <アドレス>を省略した場合、現在のPCから<ステップ数>分実行します。

Examples

```
#U,100,5↵
 *PC=01AA IR=ADI OR   A,B   A=5 B=0 X=04F T=03F F=ID.C SP=13

#U,,1↵
 *PC=01AB IR=FFF NOP7       A=5 B=0 X=04F Y=03F F=ID.C SP=13

#
```

Notes

(1) Uコマンドはリアルタイム性がありません。このためターゲットのタイマー等が実際より多く更新される場合があります。(詳細は2.3.13項を参照してください。)

(2) 本コマンド実行後のHコマンド入力では* NO HISTORY DATA *が表示されます。したがって、ヒストリ解析を行う場合はGコマンドを使用してください。

BREAK ENABLE MODE SET/BREAK DISABLE & SYNC MODE SET BE, BSYN

Format

#BE↵
#BSYN↵

Function

ブレークイネーブルモード、ブレークディセーブルモードを設定します。

(1) BE … ブレークイネーブルモードに設定します。BA、BD、BRまたはBM条件にEvaluation Board CPUの状態が一致するとブレークします。

(2) BSYN … ブレークディセーブル(シンクロナス)モードに設定します。
BA、BR、BDまたはBM条件にEvaluation Board CPUの状態が一致するとICEのSYNC端子にパルスを出力し、ブレークはしません。

(3) 電源投入時はブレークイネーブルモードになっています。

Examples

```
#BE↵
 * BREAK ENABLE MODE

#BSYN↵
 * BREAK DISABLE MODE
 * BREAK STOP MODE

#
```

Note

ブレークイネーブル/ディセーブルについては2.3.2項を参照してください。

BREAK TRACE MODE SET BT

Format

#BT↵ **(トグル)**

Function

ブレークストップモードまたはブレークトレースモードを選択します。

(1) コマンド入力ごとに指定が反転します。

(2) 電源投入時はブレークストップモードとなっています。

Examples

```
#BT↵
 * BREAK TRACE MODE   ..... 電源投入時STOPモードになっているため、コマンド入力
 * BREAK ENABLE MODE      でTRACEモードとなります。

#BT↵
 * BREAK STOP MODE    ..... コマンド入力で指定が反転します。

#
```

Note

ブレークストップ/トレースモードについては2.3.2項を参照してください。

BRKSEL *BREAK ADDRESS MODE SELECT*

Format

#BRKSEL,REM↵
#BRKSEL,CLR↵

Function

BA(ブレークアドレス条件)指定によってブレークした場合に、以前設定されていたBA条件をそのまま残すか、解除するかを指定します。

(1) 電源オン時にはクリアモード(CLRモード)となっています。

(2) 複数のブレーク条件を設定し次々とブレークポイントまでGOさせる場合は、BA条件を残すリメインモード(REMモード)が便利です。ブレークポイントをブレークごとに変えてデバッグする場合はBA条件クリアモード(CLRモード)が便利です。

Examples

```
#BA,0100↵

#BRKSEL,REM↵                          ..... リメインモードに設定します。

#BC↵
 BA 0100
    :

#G↵
 *PC=100
 *EMULATION END STATUS = BREAK HIT ..... ブレーク条件にヒットしブレークします。
 *RUN TIME = 10mS

#BA,200↵                              ..... 新たなブレーク条件を設定します。

#BC↵
 BA 0100                              ..... ブレーク前の条件は残ったままです。
 BA 0200
    :

#BRKSEL,CLR↵                          ..... クリアモードにします。

#G↵
 *PC=101
 *EMULATION END STATUS = BREAK HIT ..... ブレーク条件にヒットします。
 *RUN TIME = 30mS

#BA,300↵                              ..... ブレーク条件を新たに設定します。

#BC↵
 BA 0300                              ..... ブレーク前の条件は解除されます。
    :

#BA,350,3A0↵

#BC↵
 BA 0300
 BA 0350                              ..... ブレーク後の条件は残ります。
 BA 03A0

#
```

2.4.4 FILEコマンド群

RF READ PROGRAM FILE VIII-64
RFD READ DATA FILE VIII-64
VF VERIFY PROGRAM FILE VIII-65
VFD VERIFY DATA FILE VIII-65
WF WRITE PROGRAM FILE VIII-66
WFD WRITE DATA FILE VIII-66
CL CONDITION LOAD VIII-67
CS CONDITION SAVE VIII-67
OPTLD READ HEXA DATA FILE VIII-68

RF, RFD *READ PROGRAM/DATA FILE*

Format

#RF,<ファイル名>↵
#RFD,<ファイル名>↵

Function

ファイルをエミュレーションメモリ上にロードします。

(1) RF … <ファイル名>で示されるHEXファイルをエミュレーションプログラムメモリにロードします。

(2) RFD … <ファイル名>で示されるHEXファイル(データRAM)をデータメモリにロードします。

Examples

```
#RF,C6200A0↵          ..... C6200A0H.HEXとC6200A0L.HEXファイルをプログラムメモリにロードします。
#RFD,WORK↵            ..... WORKD.HEXファイルをデータメモリにロードします。

#
```

Notes

(1) メモリエリア(プログラムメモリ…3FF、データメモリ…7E、S1C6S3N7/6S3B7/6S3L7の場合)を越える領域へのロード、またはFDのファイルのフォーマットエラーを検出した場合
`* FILE DATA FORMAT ERROR *`を表示しコマンドを終了します。
このときエミュレーションプログラムメモリ、データメモリの内容は保証されません。

(2) データメモリへはI/O メモリ、セグメントメモリ、未使用領域はロードされません。

(3) ファイルフォーマットはHEXフォーマットとします。
(詳細は2.7項を参照してください。)

(4) ファイル形式はS1C62XXX Cross Assemblerで作成した形式とします。
(詳細は"III Cross Assembler ASM62XX"を参照してください。)

(5) コマンド実行中ESCキーは無効となります。

(6) ホスト側の入出力エラー(ドライブなし、FDのエラー等)が検出された場合、OSの処理にまかせます。したがって、ICS62XXが終了する場合があります。

(7) 未定義命令を検出するとエラー表示を行い、ICS62XXプログラムを終了します。
(詳細は2.5項を参照してください。)

VERIFY PROGRAM/DATA FILE VF, VFD

Format

#VF,<ファイル名>↵
#VFD,<ファイル名>↵

Function

エミュレーションメモリの内容とファイルの内容を比較します。

(1) VF … エミュレーションプログラムメモリ内容と<ファイル名>で指定されたHEXファイルの内容を比較します。

(2) VFD … エミュレーションデータメモリ内容(データRAM)と<ファイル名>で指定されたHEXファイルの内容を比較します。

Examples

```
#VF,C6200A0↵
 ADDR  FD:ICE
 0100 FFF:FFC
 0300 FFC:FFB

#VFD,DATA↵
 ADDR  FD:ICE
  001   1:3
 * ESC *

#
```

..... C6200A0H.HEXとC6200A0L.HEXファイルとプログラムメモリの内容を比較します。

..... 不一致が検出された場合のみアドレスとFD内容、メモリ内容を表示します。

..... ESCキーにより表示を中断し、コマンドを終了します。(エラー表示中のみ)

Notes

(1) VIII-64の注意事項(1)、(3)、(4)、(6)が本コマンドでも適用されます。

(2) エラー表示中のみESCキー入力は有効で、コマンドを終了します。

(3) データメモリ中のI/O メモリ、セグメントメモリ、未使用領域は比較されません。

WF, WFD *WRITE PROGRAM/DATA FILE*

Format

#WF,<ファイル名>↵
#WFD,<ファイル名>↵

Function

エミュレーションメモリの内容をファイルにセーブします。

(1) WF … エミュレーションプログラムメモリの内容を<ファイル名>で指定するファイルにセーブします。

(2) WFD … エミュレーションデータメモリの内容(データRAM)を<ファイル名>で指定するファイルにセーブします。

Examples

```
#WF,C6200A0↵          ..... プログラムメモリの内容をC6200A0H.HEXとC6200A0L.HEX
                            ファイルにセーブします。
#WFD,WORK↵            ..... データメモリの内容をWORKD.HEXファイルにセーブします。

#WF,ABCDEFGH↵
 * COMMAND ERROR *    ..... ファイル名が8文字以上の場合はエラーとなります。

#
```

Notes

(1) VIII-64の注意事項(3)、(4)、(5)、(6)が本コマンドでも適用されます。

(2) データメモリ中のI/O メモリ、セグメントメモリ、未使用領域はセーブされません。

CONDITION LOAD/SAVE CL, CS

Format

#CL,<ファイル名>↵
#CS,<ファイル名>↵

Function

ICEのエミュレーションメモリ内容と各種設定内容をファイルからロード、またはファイルにセーブします。

(1) CL … <ファイル名>で指定したファイルからプログラム、データをエミュレーションプログラムメモリ、データメモリにそれぞれロードします。また、各種コマンド設定条件をロードします。

(2) CS … 現在のICEのエミュレーションプログラムメモリ、データメモリの内容および各種コマンドの設定条件(ブレークの設定状態など)を<ファイル名>で指定するファイルにセーブします。

(3) ロード、セーブする内容は以下のとおりです。

- ターゲットプログラム(エミュレーションプログラム)
- ターゲットデータ(エミュレーションデータ)
- 現在のEvaluation Board CPUのレジスタ値(A、B、X、Y、F、SP、PC)
- 現在のブレーク指定情報(BA、BD、BRまたはBMコマンドの設定条件)
- ブレークモードの情報(実行時間/ステップ、ブレークストップ/ブレークトレース、ブレークイネーブル/ブレークSYNC、オンザフライあり/なし)

(4) 本コマンドはICEの電源をオフ後、同じ条件で再起動する場合に有効です。

Examples

```
#CS,TEST↵      ..... 現在のICEの設定条件をTESTC.HEXファイルにセーブします。
   :                 エミュレーションプログラムメモリの内容はTESTH.HEXとL.HEX
電源オフ             に、データメモリの内容はTESTD.HEXにセーブされます。
電源オン
   :

#CL,TEST↵      ..... CSでセーブした内容をロードします。
                     ICEは電源オフ前の状態に復帰します。
#
```

Notes

(1) VIII-64の注意事項(1)、(2)、(3)、(4)、(5)、(6)が本コマンドでも適用されます。

(2) CS,<ファイル名>で指定できるファイル名は7文字以内です。

OPTLD *READ HEXA DATA FILE*

Format

#OPTLD,0,<ファイル名>↵

Function

メロディHEXファイルの内容をEvaluation Boardのメロディデータメモリ上にロードします。対象となるファイルは、メロディアセンブラの出力するHEXファイルで、ファイルの形式は、インテルHEXフォーマットです。

Example

```
#OPTLD,0,C2XXYYY↵      ..... C2XXYYY.HEXファイルをメロディデータメモリに
                             ロードします。
#
```

2.4.5 ROMコマンド群

RP LOAD ROM PROGRAM VIII-70
VP VERIFY ROM PROGRAM VIII-71
ROM ROM TYPE SELECT VIII-72

RP *LOAD ROM PROGRAM*

Format

#RP↵

Function

ICEのROMソケット(LOW、HIGH)上のROMから、エミュレーションプログラムメモリにプログラムをロードします。ROMのデータ内容がすべてFFHの場合はROM未実装とみなします。

Examples

```
#RP↵
 * NO ROM H/L *      ..... LOW、HIGH共にROMが未実装のためエラーとなります。

#RP↵
 * NO ROM H *        ..... HIGH側ROMが未実装のためエラーとなります。

#RP↵
                     ..... ROM内容が正しくロードされました。
#
```

Notes

(1) ロード有効範囲についてはROMコマンド(VIII-72)を参照してください。

(2) 未定義コード検出時はICS62XXプログラムを終了しOSに制御を戻します。
(2.5項を参照してください。)

VERIFY ROM PROGRAM VP

Format

#VP↵

Function

ICEのROMソケット(LOW、HIGH)上のROM内容とエミュレーションプログラムメモリの内容を比較します。
データ内容に不一致がある場合、そのアドレスとデータ内容を表示します。

Examples

```
#VP↵

#                          ..... 比較結果が正しい場合は次のコマンド待ちになります。
 :
 :

#VP↵
 ADDR ROM:ICE
 0100 FFF:FFC              ..... 不一致情報(ROMアドレス、ROM内容、ICEエミュレーション
 0300 0FF:0FC                    メモリ内容)をすべて表示します。
  :    :   :
 03FF 000:001

#VP↵
 * NO ROM H *              ..... HIGH側ROMが未実装のためエラーとなります。

#VP↵
 ADDR ROM:ICE
 0100 FFF:FFC
 0300 0FF:0FC
  :    :   :
 * ESC *                   ..... [ESC]入力により表示を中断し、コマンドを終了します。

#
```

ROM *ROM TYPE SELECT*

Format

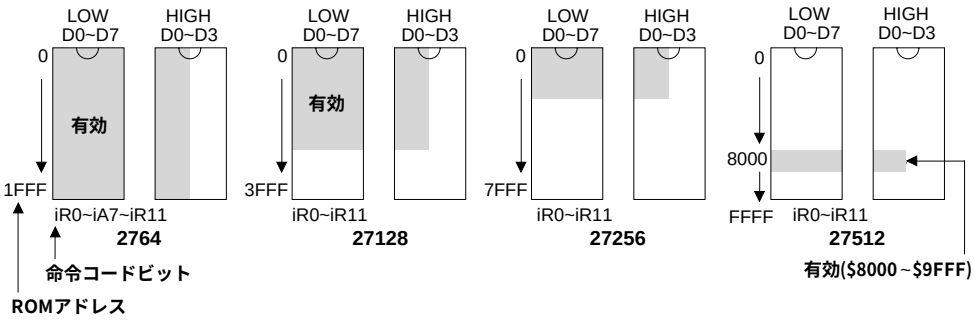
#ROM↵ **(ガイダンス付)**

Function

ICEのROMソケット(LOW、HIGH)に実装するROMのタイプを指定します。

(1) 2764、27128、27256、27512の選択が可能です。

(2) ROMのタイプによりICEにロードされる領域は以下のとおりです。



Examples

```
#ROM↵
 *ROM 64:↵        ..... 初期値は64指定となっています。
                        ↵入力のみでは変更なしとみなし、次のコマンド待ちとなります。
#ROM↵
 *ROM 64:256↵     ..... 27256指定に変更します。

#ROM↵
 *ROM 256:FF↵     ..... 64、128、256、512以外の指定はエラーとなります。

#ROM↵
 *ROM 256:↵

#
```

Note

LOW、HIGHのICソケットに実装するROMは同一タイプのものにしてください。

2.4.6 CONTROLコマンド群

I	INITIALIZE TARGET CPU	VIII-74
TIM	TIME OR STEP MODE SELECTION	VIII-74
OTF	ON THE FLY MODE SET	VIII-75
Q	QUIT	VIII-76

I　　*INITIALIZE TARGET CPU*

Format

#I↵

Function

Evaluation Board CPUをリセットします。
ICEの設定条件(ブレーク等)は不変です。

Example

```
#I↵

#                          次のコマンド待ちになります。
```

TIM　　*TIME OR STEP MODE SELECTION*

Format

#TIM↵　　**(トグル)**

Function

ターゲットプログラム実行時の実行時間測定カウンタを実時間測定モード、またはステップ数測定モードにします。

(1) 電源投入時は実時間測定モードとなっています。

(2) コマンド入力ごとに指定が反転します。

Examples

```
#TIM↵
 * STEP COUNT MODE    ..... 電源投入時はタイムカウントモードのためコマンド入力で
                            ステップモードとなります。
#TIM↵
 * TIME COUNT MODE    ..... コマンド入力ごとに指定を反転します。

#
```

Note

時間測定/ステップ数測定については2.3.10項を参照してください。

ON THE FLY MODE SET OTF

Format

#OTF↲ **(トグル)**

Function

ターゲットプログラム実行時のオンザフライ表示を行うか行わないかを選択します。

(1) コマンド入力ごとに指定が反転します。

(2) 電源投入時はオンザフライ表示モードとなっています。

(3) ホストにプリンタを接続している場合は表示オフモードにすると便利です。

Examples

```
#OTF↲
 * ON THE FLY OFF        ..... 電源投入時は表示モードとなっているため、コマンド入力
                               で表示オフモードとなります。
#OTF↲
 * ON THE FLY ON         ..... オンザフライ表示モードとなります。

#G↲
 * PC=xxxx               ..... Evaluation Board CPUの実行PCを一定周期で表示します。
   :
   :
#OTF↲
 * ON THE FLY OFF

#G↲
                               ..... PC表示は行いません。
#
```

Note

オンザフライ機能については2.3.4項を参照してください。

Q *QUIT*

Format

#Q↵

Function

ICS62XXプログラムを終了しOSに制御を戻します。

Example

```
#Q↵
B>                          ..... パソコンOSのコマンド待ちになります。
B>ICS62XX↵                  ..... 再度ICEを起動します。
 …エプソンロゴを約1秒表示します…
 * ICE POWER ON RESET *
 * DIAGNOSTIC TEST OK *

#                           ..... ICEのコマンド待ちになります。
```

2.4.7 HELPコマンド

HELP

Format

#HELP↵ **(ガイダンス付)**

#HELP,n↵ (n=1 to 8)

Function

ICEのコマンドを表示します。

(1) オプション(,n)指定のない場合、全コマンドのタイトルを一画面に表示します。

(2) オプション(,n)指定を行った場合、nはコマンド属性を示します。同一属性のコマンドの説明表示を行います。

nの値	属　性
1	DISPLAY COMMAND
2	SET COMMAND
3	BREAK & GO COMMAND
4	FILE COMMAND
5	ROM COMMAND
6	CONTROL COMMAND
7	ALL COMMAND DISPLAY
8	BASIC COMMAND DISPLAY

Examples

```
#HELP↵
```

次頁のメッセージを参照してください。

```
KEY IN 1.8 ENTER OR ENTER ONLY :1↵
```

DISPLAY COMMANDが表示されます。
(次頁を参照してください。)

```
#HELP,F↵                    ..... 1～8以外の指定はエラーとなります。
 * COMMAND ERROR *

#
```

HELP

Format

#HELP↵ **(ガイダンス付)**

#HELP,n↵ (n=1 to 8)

Examples

```
#HELP↵
 1.DISPLAY COMMAND        #L   #DP  #DD  #DR  #H   #HB  #HG  #HS  #HSW #HSR
                          #HP  #CHK #DXY #CVD #HAD
 2.SET COMMAND            #A   #FP  #FD  #MP  #MD  #SP  #SD  #SR  #SXY #HC
                          #HA  #HAR #HPS #CVR
 3.BREAK and GO COMMAND   #BA  #BD  #BR  #BM  #BAR #BDR #BRR #BMR #BRES
                          #BC  #G   #T   #U   #BSYN #BE #BT  #BRKSEL
 4.FILE COMMAND           #RF  #VF  #WF  #RFD #VFD #WFD #CL  #CS  #OPTLD
 5.ROM COMMAND            #RP  #VP  #ROM
 6.CONTROL COMMAND        #I   #TIM #OTF #Q
 7.ALL COMMAND DISPLAY
 8.BASIC COMMAND DISPLAY

 KEY IN 1..8 ENTER or ENTER ONLY :↵

#

#HELP,1↵
 1.DISPLAY COMMAND
 (1)#L,addr1,addr2   program code and mnemonic display.
 (2)#DP,addr1,addr2  program area HEX display.
 (3)#DD,addr1,addr2  data area HEX display.
 (4)#DR              register data display.
 (5)#H,addr1,addr2   history data display.
 (6)#HB or #HG       history data display BACK or GO NEXT.
 (7)#HS,addr         history serch and display.
 (8)#HSW,addr        memory write history serch and display.
 (9)#HSR,addr        memory read history serch and display.
(10)#HP              current history pointer display.
(11)#CHK             ice initial self test information display.
(12)#DXY             X,Y register and MX,MY data display.
(13)#CVD,addr1,addr2 coverage area display.
(14)#HAD             history PC area information display.

#

#HELP,2↵
 2.SET COMMAND
 (1)#A,addr                 assemble program.
 (2)#FP,addr1,addr2,data    fill program addr1 to addr2 by data.
 (3)#FD,addr1,addr2,data    fill data addr1 to addr2 by data.
 (4)#MP,addr1,addr2,addr3   move program from addr1..addr2 to addr3.
 (5)#MD,addr1,addr2,addr3   move data from addr1..addr2 to addr3.
 (6)#SP,addr                program area patch.
 (7)#SD,addr                data area patch.
 (8)#SR or #SR,reg,data     register patch.
 (9)#SXY                    MX,MY patch.
(10)#HC,S/C/E               history Start/Center/End set.
(11)#HA,addr1,addr2         set PC addr1..addr2 save to history memory.
     (#HA,ALL)              (all data save.)
(12)#HAR,addr1,addr2        inhibit PC addr1..addr2 save to history memory.
     (#HAR,ALL)             (all reset.)
(13)#HPS,addr               set history pointer.
(14)#CVR                    reset coverage information.

#
```

HELP

Format

#HELP↵ **(ガイダンス付)**

#HELP,n↵ (n=1 to 8)

Examples

```
#HELP,3↵
 3.BREAK and GO COMMAND
 (1)#BA,addr,...    set break address.
 (2)#BD             set break data condition.
 (3)#BR             set break register condition.
 (4)#BM             set break address,data,register multiple condition.
 (5)#BAR            reset break address.
 (6)#BDR            reset break data condition.
 (7)#BRR            reset break register condition.
 (8)#BMR            reset break address,data,register multiple condition.
 (9)#BRES           reset all break condition.
(10)#BC             break condition display.
(11)#G or #G,addr   GO current address or GO from set addr.
(12)#G,R            GO after reset cpu.
(13)#T,addr,step    single step run and display break information.
(14)#U,addr,step    single step run in ICE. and display last break information.
(15)#BSYN           set break disable mode.
(16)#BE             set break enable mode.
(17)#BT             set and reset break trace made. (alternate)
(18)#BRKSEL,CLR/REM set break address clear mode or remain mode.
#

#HELP,4↵
 4.FILE COMMAND
 (1)#RF,file                program load.
 (2)#VF,file                program verify.
 (3)#WF,file                program save.
 (4)#RFD,file               RAM data load.
 (5)#VFD,file               RAM data verity.
 (6)#WFD,file               RAM data save.
 (7)#CL,file                program,RAM data,break condition load.
 (8)#CS,file                program,RAM data,break condition save.
 (9)#OPTLD,option no.,file  HEXA data load.
#

#HELP,5↵
 5.ROM COMMAND
 (1)#RP       program load from ROM.
 (2)#VP       program verify ice:ROM.
 (3)#ROM      ROM type select. (64,128,256,512)
#

#HELP,6↵
 6.CONTROL COMMAND
 (1)#I        reset target CPU.
 (2)#TIM      set step count mode or time count mode. (alternate)
 (3)#OTF      set on-the-fly display mode or inhibit mode. (alternate)
 (4)#Q        program exit.
#
```

HELP

Format

#HELP↵ **(ガイダンス付)**

#HELP,n↵ (n=1 to 8)

Examples

```
#HELP,8↵
 8.BASIC COMMAND
 (1)#L,addr1,addr2   program code and mnemonic display.
 (2)#DD,addr1,addr2  data area HEX display.
 (3)#DR              register data display.
 (4)#BC              break condition display.
 (5)#H,addr1,addr2   history data display.
 (6)#A,addr          assemble program.
 (7)#SP,addr         program area patch.
 (8)#SD,addr         data area patch.
 (9)#SR              register patch.
(10)#BA,addr,...     set break address.
(11)#BD              set break data condition.
(12)#BR              set break register condition.
(13)#BM              set break address,data,register multiple condition.
(14)#BRES            reset all break condition.
(15)#G or #G,addr    GO current address or GO from set address.
(16)#T,addr,step     single step run and display break information.
(17)#CL,file         program,RAM data,break condition load.
(18)#CS,file         program,RAM data,break condition save.
(19)#I               reset target CPU.
(20)#Q               program exit.

#
```

2.5 エラーメッセージ一覧

以下にICS62XXのエラーメッセージを示します。

エラーメッセージ *** COMMUNICATION ERROR OR ICE NOT READY ***
意味 ICEが未接続、またはICEが電源オフ
回復方法 ホストの電源をオフし、ケーブル接続して再度起動させる。またはICEの電源をオンにする。

エラーメッセージ *** TARGET DOWN(1) ***
意味 Evaluation Boardが未接続 (ICEのパワーオン時に判定)
回復方法 ICEの電源をオフし、Evaluation Boardを接続する。その後ICEの電源をオンにする。

エラーメッセージ *** TARGET DOWN(2) ***
意味 Evaluation Boardが未接続 (コマンド実行時に判定)
回復方法 ICEの電源をオフし、Evaluation Boardを接続する。その後ICEの電源をオンにする。

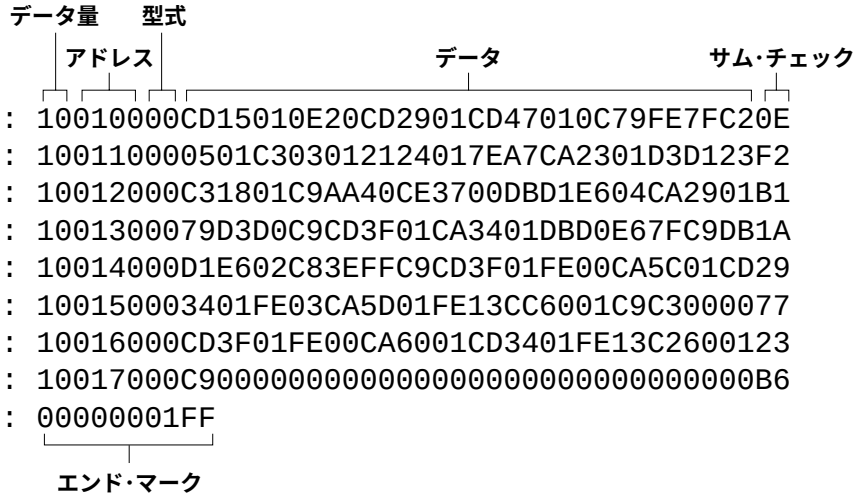
エラーメッセージ *** UNDEFINED PROGRAM CODE EXIST ***
意味 ROMやFDからロードしたプロ グラム内に未定義コードが検出された。(ICS62XXは終了)
回復方法 ROM、FDをS1C62XXX Cross Assemblerで作成したものに交換して再度ICEを起動させる。

エラーメッセージ *** COMMAND ERROR ***
意味 コマンド入力で指定ミスがあった。
回復方法 正しいコマンドを再度入力する。

エラー **電源オン後無応答**
意味 ICE～HOSTケーブルのHOST側が接続されていない。
回復方法 ケーブルを接続する。

2.6 HEXファイルフォーマット

HEXフォーマット例



- **データ量 (1バイト)** : データの領域に含まれているデータの数量を表しており、最大10H (16個)です。
- **アドレス (2バイト)** : 各行の先頭データの存在するアドレスを示しています。
- **型　　式 (1バイト)** : ヘキサ･フォーマットの型式(タイプ)を示すもので、現在は"00"のみです。
- **データ (MAX 16バイト)** : データを、ヘキサ形式で表しています。
- **サム･チェック (1バイト)** : "データ量のバイト"から"データの最後のバイト"までの 全バイトの内容を加算し、その加算結果の2の補数を16進で表しています。
- **エンド･マーク** : HEXファイルの終了の印として必要なものです。

IX

Mask Data Checker

MDC62XX

ここでは4-bit Single Chip Microcomputer S1C62XXX用
マスクデータチェッカMDC62XXの操作方法を中心に
解説を行っています。

Mask Data Checker MDC62XX

- 目 次 -

1 機種による相違点 IX-1

2 MDC62XXの概要 IX-1

2.1 概要 IX-1

2.2 実行フローと入出力ファイル IX-1

3 MDC62XXの操作方法 IX-2

3.1 データファイルの準備 IX-2

3.2 MDC62XXの実行 IX-2

3.2.1 MDC62XXの起動 IX-2

3.2.2 データのパック IX-3

3.2.3 データのアンパック IX-3

4 エラーメッセージ IX-4

4.1 データエラー IX-4

4.1.1 プログラムデータエラー IX-4

4.1.2 ファンクションオプションデータエラー IX-4

4.1.3 セグメントオプションデータエラー IX-4

4.2 ファイルエラー IX-4

4.3 システムエラー IX-4

5 パックファイルの構成 IX-5

5.1 プログラムデータ、メロディROMデータ、スケールROMデータ IX-6

5.2 セグメントデータ IX-6

1 機種による相違点

MDC62XXの入出力ファイル、およびそのファイルを作成するプログラムの中で、以下の2種類については、機種により存在しない場合があります。

(1) SOG62XXおよびC2XXYYYS.DOCはセグメントオプションを持つ機種にのみ設定されています。

(2) MLA628XおよびC28XYYYA.DOCはメロディデータを持つ機種にのみ設定されています。

上記の機能を持たない機種を使用される場合は、以下に記載のそれぞれのプログラム名およびデータファイル名を無視してください。

2 MDC62XXの概要

2.1 概要

マスクデータチェッカMDC62XXは、お客さまの作成されたプログラムデータ(C2XXYYYH.HEX、C2XXYYYL.HEX)、オプションデータ(C2XXYYYF.DOC、C2XXYYYS.DOC)およびメロディデータ(C28XYYYA.DOC)をチェックし、マスクパターン生成のためのデータファイル(C62XXYYY.PAn)を作成するソフトウェアツールです。お客さまは、このデータファイルをセイコーエプソンに提出してください。

また、MDC62XXには生成されたデータファイル(C62XXYYY.PAn)を元のファイル形式(C2XXYYYH.HEX、C2XXYYYL.HEX、C2XXYYYF.DOC、C2XXYYYS.DOC、C28XYYYA.DOC)に復元する機能もあります。

2.2 実行フローと入出力ファイル

図2.2.1にMDC62XXの実行フローを示します。

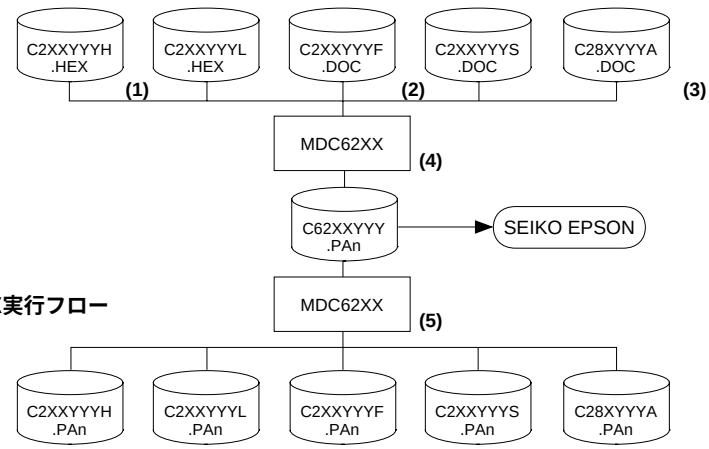


図2.2.1 MDC62XX実行フロー

(1) プログラムデータファイル(C2XXYYYH.HEX、C2XXYYYL.HEX)の用意
クロスアセンブラ(ASM62XX)から生成されたプログラムデータファイルを用意します。

(2) オプションデータファイル(C2XXYYYF.DOC、C2XXYYYS.DOC)の用意
オプションジェネレータ(FOG62XX、SOG62XX)から生成されたオプションデータファイル(ファンクションオプション、セグメントオプション)を用意します。

(3) メロディデータファイル(C28XYYYA.DOC)の用意
メロディアセンブラ(MLA628X)から生成されたメロディデータファイルを用意します。

(4) データのパック
MDC62XXを用いて、プログラムデータファイル、オプションデータファイル、およびメロディデータファイルを1つのマスク用データファイル(C62XXYYY.PAn)にします。このファイルをセイコーエプソンに提出していただきます。

(5) データのアンパック
マスク用データファイル(C62XXYYY.PAn)は、MDC62XXを用いて元のプログラムデータファイル、オプションデータファイル、およびメロディデータファイルに復元することができます。

3 MDC62XXの操作方法

3.1 データファイルの準備

セイコーエプソンにデータを提出される場合は、クロスアセンブラ(ASM62XX)、ファンクションオプションジェネレータ(FOG62XX)、セグメントオプションジェネレータ(SOG62XX)およびメロディアセンブラ(MLA628X)から生成されたデバッグ済みのデータをMDC62XX.EXEと同じディスク(ディレクトリ)に用意します。
各ファイル名は、必ず以下のとおりとしてください。(YYYの部分は、セイコーエプソンより提示される内容としてください。)

- **プログラムデータ** (HIGH側) C2XXYYYH.HEX
  (LOW側) C2XXYYYL.HEX
- **オプションデータ** (ファンクションオプション) C2XXYYYF.DOC
  (セグメントオプション) C2XXYYYS.DOC
- **メロディデータ** (メロディROM、スケールROM、メロディオプション) ... C28XYYYA.DOC

3.2 MDC62XXの実行

3.2.1 MDC62XXの起動

MDC62XXを起動させるには、DOSのコマンドレベル(A>などのプロンプトが表示されている状態)で次のプログラム名を入力します。DMS6200から起動させる場合は、メニュー画面で"MDC62XX.EXE"を選択してください。

A>MDC62XX↵	**↵はリターンキーの入力を表します。**

MDC62XXが起動すると、次のメッセージが表示されます。

```
                *** E0C62XX PACK / UNPACK PROGRAM Ver 1.00 ***

     EEEEEEEEEE   PPPPPPPP        SSSSSSS       OOOOOOOO     NNN      NNN
     EEEEEEEEEE   PPPPPPPPPP     SSS   SSSS    OOO    OOO    NNNN     NNN
     EEE          PPP     PPP   SSS     SSS   OOO      OOO   NNNNN    NNN
     EEE          PPP     PPP    SSS          OOO      OOO   NNNNNN   NNN
     EEEEEEEEEE   PPPPPPPPPP      SSSSSS      OOO      OOO   NNN NNN NNN
     EEEEEEEEEE   PPPPPPPP           SSSS     OOO      OOO   NNN  NNNNNN
     EEE          PPP                 SSS    OOO      OOO   NNN   NNNNN
     EEE          PPP           SSS     SSS   OOO      OOO   NNN    NNNN
     EEEEEEEEEE   PPP           SSSS   SSS     OOO    OOO    NNN     NNN
     EEEEEEEEEE   PPP             SSSSSSS       OOOOOOOO     NNN      NN

                  (C) COPYRIGHT 1991 SEIKO EPSON CORPORATION

                         --- OPERATION MENU ---

                                1. PACK
                                2. UNPACK

                         PLEASE SELECT NO.?
```

ここでオペレーション内容を問い合わせてきます。提出されるマスク用データを作成する場合は1を選択し、マスク用データを分割して元のファイル形式(C2XXYYYH.HEX、C2XXYYYL.HEX、C2XXYYYF.DOC、C2XXYYYS.DOC、C28XYYYA.DOC)に復元する場合は2を選択します。

3.2.2 データのパック

セイコーエプソンに提出するデータを生成する場合は、3.2.1で1を選択します。
1の入力後、以下のように生成ファイル名を問い合わせてきます。

```
   C2XXYYYH.HEX --------+
                        |
   C2XXYYYL.HEX --------+
                        |
   C2XXYYYF.DOC --------+-------- C62XXYYY.PAn (PACK FILE)
                        |
   C2XXYYYS.DOC --------+
                        |
   C28XYYYA.DOC --------+

PLEASE INPUT PACK FILE NAME (C62XXYYY.PAn) ? C62XXYYY.PA0↵
```

YYYの部分はセイコーエプソンより提示される内容としてください。また、セイコーエプソンにデータを提出された後、プログラム等の不具合によりデータの再提出される場合は、nの数値を1つ増やして入力します。
(例. "C62XXYYY.PA0"を提出された後、再提出する場合は、"C62XXYYY.PA1"とします。)

データのパックを行う場合は、実行前にプログラムデータファイルおよびオプションデータファイルを作成しておく必要があります。

ファイル名の入力が終了するとマスク用データ生成し、それぞれのファイル名を表示します。

```
   C2XXYYYH.HEX --------+
                        |
   C2XXYYYL.HEX --------+
                        |
   C2XXYYYF.DOC --------+-------- C62XXYYY.PA0
                        |
   C2XXYYYS.DOC --------+
                        |
   C28XYYYA.DOC --------+
```

これでマスク用ファイル(C62XXYYY.PAn)が生成されました。このファイルをセイコーエプソンに提出してください。

**注: プログラムデータとしてはクロスアセンブラ(ASM62XX)の"-N"オプションをつけずに生成されたものを使用してください。クロスアセンブラの"-N"オプションを用いて生成されたプログラムデータをパックすると、未定義となるプログラム領域にFFHのコードがつめられます。
この場合、次のメッセージが表示されます。**

```
    WARNING: FILLED <file_name> FILE WITH FFH.
```

3.2.3 データのアンパック

パックデータを元のファイル形式に復元する場合は、3.2.1で2を選択します。
2の入力後、以下のように入力データファイル名を問い合わせてきます。

```
PLEASE INPUT PACKED FILE NAME (C62XXYYY.PAn) ? C62XXYYY.PA0↵
```

ファイル名の入力が終了すると、アンパック処理を開始し、それぞれのファイル名を表示します。

```
                        +-------- C2XXYYYH.PA0
                        |
                        +-------- C2XXYYYL.PA0
                        |
   C62XXYYY.PA0 --------+-------- C2XXYYYF.PA0
                        |
                        +-------- C2XXYYYS.PA0
                        |
                        +-------- C28XYYYA.PA0
```

これでマスク用データファイル(C62XXYYY.PAn)は、元のファイル形式に復元されました。元データとの比較を行うことが可能となります。

復元された各データのファイル名は、以下のようになります。

- **プログラムデータ** (HIGH側) C2XXYYYH.PAn
  (LOW側) C2XXYYYL.PAn
- **オプションデータ** (ファンクションオプション) C2XXYYYF.PAn
  (セグメントオプション) C2XXYYYS.PAn
- **メロディデータ** (メロディROM、スケールROM、メロディオプション) ... C28XYYYA.PAn

4 エラーメッセージ

4.1 データエラー

パック動作において、プログラムデータファイル、オプションデータファイルおよびメロディデータファイルを、アンパック動作においては、パックデータファイルをそれぞれチェックし、フォーマットに異常な箇所があれば以下のようなエラーメッセージを表示します。

4.1.1 プログラムデータエラー

エラーメッセージ	説　明
1. HEX DATA ERROR : NOT COLON.	コロンが存在しない。
2. HEX DATA ERROR : DATA LENGTH. (NOT 00-20h)	1行のデータ長が"00H"から"20H"の範囲にない。
3. HEX DATA ERROR : ADDRESS.	アドレスがプログラム、メロディおよびスケールROM容量の範囲を越えている。
4. HEX DATA ERROR : RECORD TYPE. (NOT 00)	1行のレコードタイプが"00"でない。
5. HEX DATA ERROR : DATA. (NOT 00-FFh)	データが"00H"から"0FFH"の範囲にない。
6. HEX DATA ERROR : TOO MANY DATA IN ONE LINE.	1行のデータが多すぎる。
7. HEX DATA ERROR : CHECK SUM.	チェックサムが合わない。
8. HEX DATA ERROR : END MARK.	エンドマークが":00000001FF"でない。
9. HEX DATA ERROR : DUPLICATE.	同一アドレスにデータの2重定義がされている。

4.1.2 ファンクションオプションデータエラー

エラーメッセージ	説　明
1. OPTION DATA ERROR : START MARK.	スタートマークが"\OPTION"でない。(アンパック時) *
2. OPTION DATA ERROR : OPTION NUMBER.	オプションNO. がおかしい。
3. OPTION DATA ERROR : SELECT NUMBER.	オプション選択NO. がおかしい。
4. OPTION DATA ERROR : END MARK.	エンドマークが違う。(パック時は"\\END"、アンパック時は"\END"でない。) *

*** 使用するパソコンによって、"\"は"\"となる場合があります。**

4.1.3 セグメントオプションデータエラー

エラーメッセージ	説　明
1. SEGMENT DATA ERROR : START MARK.	スタートマークが"\SEGMENT"でない。(アンパック時)*
2. SEGMENT DATA ERROR : DATA.	セグメントデータがおかしい。
3. SEGMENT DATA ERROR : SEGMENT NUMBER.	SEG NO.がおかしい。
4. SEGMENT DATA ERROR : SPEC.	SEG端子の出力仕様がおかしい。
5. SEGMENT DATA ERROR : END MARK.	エンドマークが違う。(パック時は"\\END"、アンパック時は"\END"でない。) *

*** 使用するパソコンによって、"\"は"\"となる場合があります。**

4.2 ファイルエラー

エラーメッセージ	説　明
1. <File_name> FILE IS NOT FOUND.	ファイルが見つからない。または、CONFIG.SYSファイル中のファイル数設定が10以上でない。
2. PACK FILE NAME (File_name) ERROR.	パックするファイル名入力フォーマットがおかしい。
3. PACKED FILE NAME (File_name) ERROR.	アンパックするファイル名入力フォーマットがおかしい。

4.3 システムエラー

エラーメッセージ	説　明
1. DIRECTORY FULL.	ディレクトリがいっぱいである。
2. DISK WRITE ERROR.	ディスクの書き込みに失敗した。

5 パックファイルの構成

パックファイルは、以下のフォーマットにしたがって構成されています。

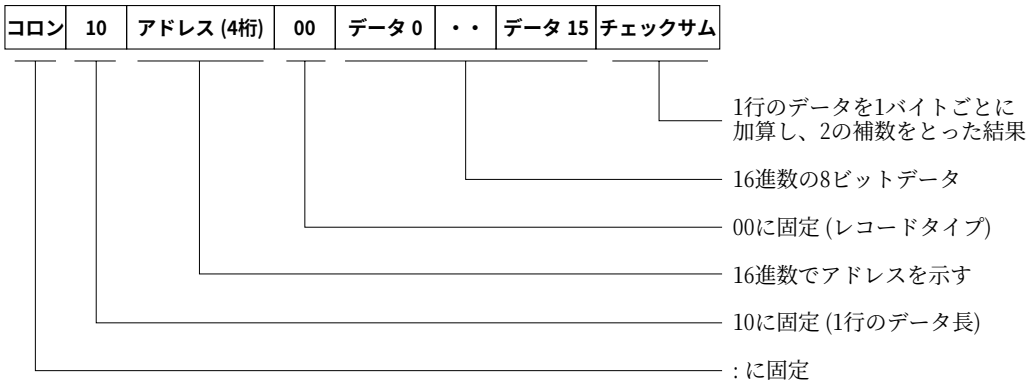
```
                                *
                                * E0C62XX MASK DATA VER 1.00
                                *
プログラムデータヘッダ          — ¥ROM1
機種名                          — E0C62XXYYY  PROGRAM ROM
                                ┌ :100000000.............................
プログラムデータ                │ :100010000.............................
HIGH側( インテルヘキサフォーマット ) │     :    :    :    :    :    :    :    :
                                └ :00000001FF
                                ┌ :100000000.............................
プログラムデータ                │ :100010000.............................
LOW側( インテルヘキサフォーマット ) │     :    :    :    :    :    :    :    :
                                └ :00000001FF
エンドマーク                    — ¥END
メロディROMヘッダ               — ¥ROM2
機種名                          — E0C628XYYY  MELODY ROM
                                ┌ :100000000.............................
メロディROMデータ               │     :    :    :    :    :    :    :    :
HIGH側( インテルヘキサフォーマット ) └ :00000001FF
                                ┌ :10000000..............................
メロディROMデータ               │     :    :    :    :    :    :    :    :
LOW側( インテルヘキサフォーマット ) └ :00000001FF
エンドマーク                    — ¥END
メロディスケールROMヘッダ       — ¥ROM3
機種名                          ┌ E0C628XYYY  SCALE ROM
メロディスケールROMデータ       │ :10000000..............................
( インテルヘキサフォーマット )  └ :00000001FF
エンドマーク                    —  ¥END
メロディオプションデータヘッダ  —  ¥OPTION1
                                ┌ *
                                │ *     OCTAVE CIRCUIT
メロディオプションデータ        │ *      32kHz ---------------------------- SELECTED
                                │  OPT2001 01
                                │     :    :    :    :    :    :    :    :
                                └  OPT2104 04
エンドマーク                    —  ¥END
ファンクションオプションヘッダ  —  ¥OPTION2
                                ┌ * E0C62XX FUNCTION OPTION DOCUMENT  VER 3.00
                                │ *
                                │ * FILE NAME    C2XXYYYF.DOC
                                │ * USER'S NAME  SEIKO EPSON CORP.
                                │ * INPUT DATE   91/07/22
ファンクションオプションデータ  │ *
                                │ * OPTION NO.1
                                │ * < DEVICE TYPE >
                                │ *     E0C62XX  ( NORMAL TYPE )  --------- SELECTED
                                │  OPT0101 01
                                └    :    :    :    :    :    :    :    :    :
エンドマーク                    — ¥END
セグメントオプションヘッダ      —  ¥SEGMENT
                                ┌ * E0C62XX SEGMENT OPTION DOCUMENT VER 3.00
                                │ *
                                │ * FILE NAME    C2XXYYYS.DOC
                                │ * USER'S NAME  SEIKO EPSON CORP.
                                │ * INPUT DATE   91/07/22
                                │ * COMMENT      TOKYO DESIGN CENTER
                                │ *              421-8 HINO HINO-SHI TOKYO 191 JAPAN
セグメントオプションデータ      │ *
                                │ *
                                │ * OPTION NO.**
                                │ *
                                │ * < LCD SEGMENT DECODE TABLE >
                                │ *
                                │ * SEG COM0 COM1 COM2 COM3
                                │ *
                                │    0  ...  ...  ...  ...  S
                                │    1  ...  ...  ...  ...  C
                                └    :   :    :    :    :
エンドマーク                    — ¥END
```

＊使用するパソコンによって、"\"は"\"となる場合があります。

5.1 プログラムデータ、メロディROMデータ、スケールROMデータ

プログラムデータ、メロディROMデータ、スケールROMデータは、下記のようなインテルヘキサフォーマットにしたがって構成されています。

■ データ行

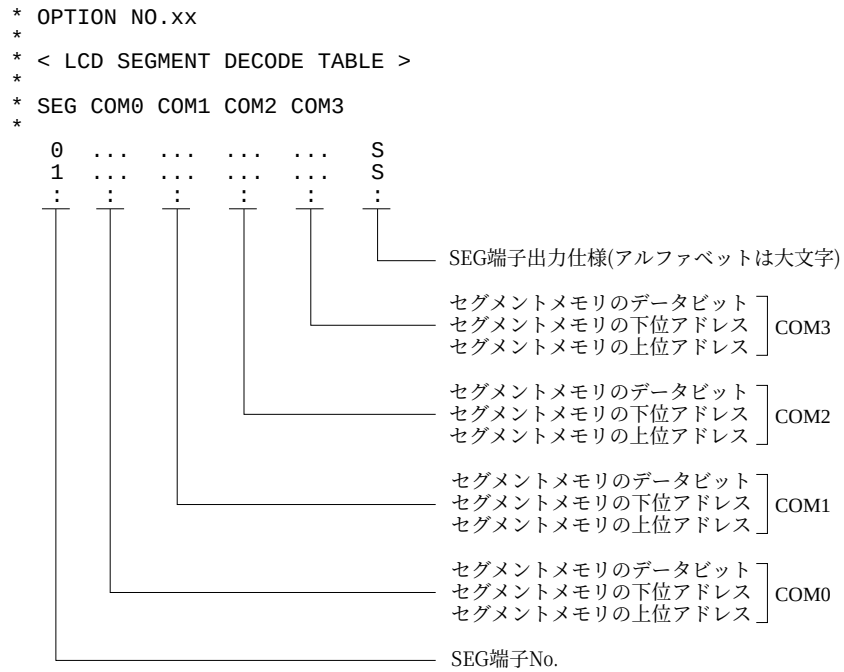


■ エンドマーク

:00000001FF

5.2 セグメントデータ

セグメントデータは、以下のようなフォーマットにしたがって構成されています。(4コモンの例)



セイコーエプソン株式会社
電子デバイス営業本部

ED営業推進部　〒191-8501　東京都日野市日野421-8
IC営業技術G　TEL (042)587-5816(直通)　FAX (042)587-5624
〈東日本〉
ED東京営業部　〒191-8501　東京都日野市日野421-8
東京IC営業G　TEL (042)587-5313(直通)　FAX (042)587-5116
〈西日本〉
ED大阪営業部　〒541-0059　大阪市中央区博労町3-5-1　エプソン大阪ビル15F
TEL (06)6120-6000(代表)　FAX (06)6120-6100
〈東海・北陸〉
ED名古屋営業部　〒461-0005　名古屋市東区東桜1-10-24　栄大野ビル4F
TEL (052)953-8031(代表)　FAX (052)953-8041
〈長野〉
ED長野営業部　〒392-8502　長野県諏訪市大和3-3-5
TEL (0266)58-8171(直通)　FAX (0266)58-9917
〈東北〉
ED仙台営業所　〒980-0013　宮城県仙台市青葉区花京院1-1-20　花京院スクエア19F
TEL (022)263-7975(代表)　FAX (022)263-7990

インターネットによる電子デバイスのご紹介　http://www.epson.co.jp/device/

2001年4月改訂
1991年9月作成 ⓂB